

GUERRERO:  
Conception and Process of a Surface

Gregory Rowland Evans

May 3, 2019

Copyright © 2019 Gregory Rowland Evans

Published by [www.gregoryrowlandevans.com](http://www.gregoryrowlandevans.com)

This paper was typeset in L<sup>A</sup>T<sub>E</sub>X and all graphics were created by the author with Lilypond.

[www.latex-project.org](http://www.latex-project.org)

[lilypond.org](http://lilypond.org)

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at [www.apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0). Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*First printing, May, 2019*

# Contents

<b>1 Analysis</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Abjad: FSC and Composition as Software Development . . . . .	2
1.2.1 Formalized Score Control . . . . .	2
1.2.2 Composition as Software Development . . . . .	4
1.3 Genesis of <i>GUERRERO</i> . . . . .	5
1.4 Initial Concepts and Materials . . . . .	6
1.5 Formal Structures in <i>GUERRERO</i> . . . . .	8
1.6 Processes in the Remaining Sections of <i>GUERRERO</i> . . . . .	11
1.7 Future Implications for My Work . . . . .	14
<b>2 Supplemental Illustrations</b>	<b>15</b>
2.1 Global Form . . . . .	15
2.2 Invocation . . . . .	17
2.3 Section A . . . . .	19
2.4 Section B . . . . .	21
2.5 Section C . . . . .	23
2.6 Section D . . . . .	25

2.7	Section E	27
2.8	Section F	29
2.9	Section G	31
2.10	Section H	33
2.11	Section I	35
2.12	Section J	37
<b>3</b>	<b>Source Code</b>	<b>39</b>
3.1	Invocation	39
3.2	Section A	77
3.3	Section B	133
3.4	Section C	190
3.5	Section D	233
3.6	Section E	285
3.7	Section F	358
3.8	Section G	431
3.9	Section H	485
3.10	Section I	523
3.11	Section J	579
<b>4</b>	<b>Score</b>	<b>629</b>

# Chapter 1

## Analysis

### 1.1 Introduction

*GUERRERO* is a composition for twenty-one saxophones ranging from Contrabass to Sopranino which was composed for the Frost Saxophone Ensemble in December of 2018. The genesis of this work was strongly related to the medium in which it was created. It was composed with the Abjad<sup>1</sup> Application Programming Interface<sup>2</sup> for Formalized Score Control<sup>3</sup> in the Python<sup>4</sup> programming language, making use of built-in tools within Abjad's framework and software written by the composer. While this analysis is intended to elucidate the construction of the work, chapter three of this document contains the source code, written in Python, of *GUERRERO* allowing for the direct perusal of the materials in question.

In the Winter of 2017, I began using OpenMusic,<sup>5</sup> a music computation software produced by IRCAM.<sup>6</sup> This program allowed me to realize many of the musical ideas that I had yet to crystallize and using it was a major stepping stone in the development of my music. However, I felt that OM

---

<sup>1</sup>[abjad.mbrsri.org](http://abjad.mbrsri.org)

<sup>2</sup>API

<sup>3</sup>FSC

<sup>4</sup>[www.python.org](http://www.python.org)

<sup>5</sup>OM

<sup>6</sup>Institut de Recherche et Coordination Acoustique/Musique

had some significant drawbacks for the workflow I was hoping to develop, which led me to look into other software. The “generative task” of a given software refers to the intended use of a program.[1] The existence of a generative task implies in what ways features within a program are intended to be deployed and what the program is good at achieving. Artists are able to use software for purposes outside of its generative task, achieving fascinating results, but I eventually knew that OM’s generative task was not sufficiently open to the kind of workflow I wanted from a computer-assisted composition program.<sup>7</sup> In the Summer of 2018, I began to study the Abjad API for FSC, developed by my friend and mentor Trevor Baća along with Víctor Adán and Josiah Wolf Oberholtzer in Python.<sup>8</sup> What began as a casual interaction with a new piece of music software rapidly became a determinant of new musical potentials and helped me to form essential ideologies in my compositional practice.

## 1.2 Abjad: FSC and Composition as Software Development

### 1.2.1 Formalized Score Control

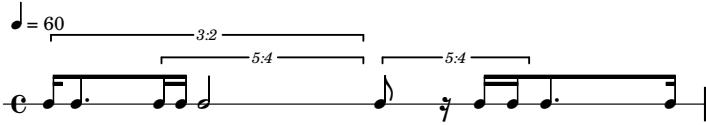
Composing with Abjad can be seen as a kind of sorcery by which numbers and data are transformed into a score to be interpreted by the performer. In fact, the software’s name is derived from the Abjad numeral system whereby letters assigned a numeric value can be used to encode words and phrases with hidden dates and other numerological symbology. There appear to be two fundamental principles held by most dedicated users of Abjad, including myself. The first principle is that of Formalized Score Control. This notion is based upon the idea that the act of composing for acoustic instruments requires a relay of some form of instruction from the composer to the interpreter. In most forms of Western classical music, the medium of transference is that of music notation in the traditional style. With FSC, the act of composition is concerned with the manipulation of notational graphemes on a page of a score that represent an action to be achieved by the interpreter, not an abstract description of what sounds may result. These graphic elements are often traditional

---

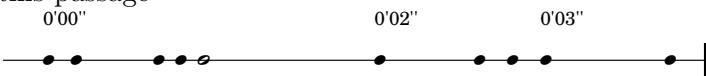
<sup>7</sup>A thorough description of the workflow I desired as well as a comparison of Abjad with similar software can be found in my masters thesis: *An Introduction to Modeling Composition through Abjad’s Model of Music Notation*

<sup>8</sup>A brief history of Abjad can be found here: [http://abjad.mbrsi.org/core\\_concepts/from\\_abjads\\_developers.html](http://abjad.mbrsi.org/core_concepts/from_abjads_developers.html)

notational figures such as note heads, beams, slurs, and more, but can also be articulations, text, lines, and other shapes. In this case, the score is a deliberate set of instructions intended to affect the mode of performance, thus the clarity, precision, and purity of this document is held as a significant priority. For instance, the difference between the notation of this passage



and this passage



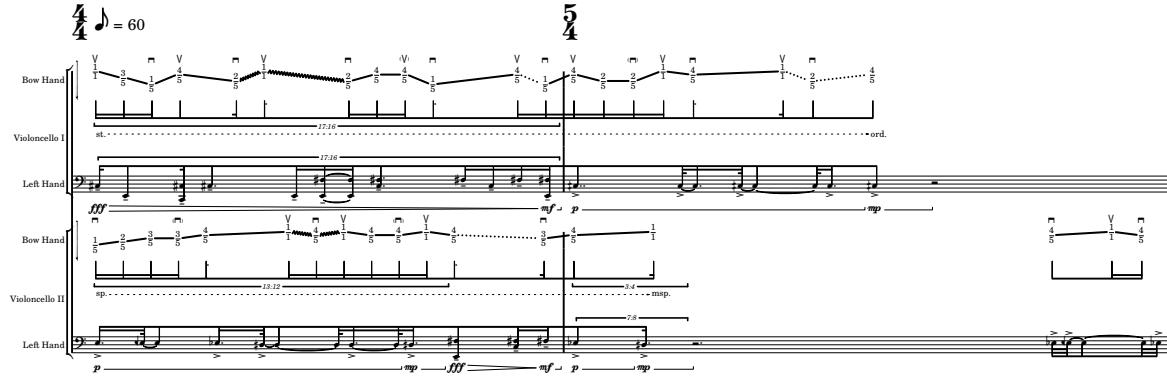
is in no way trivial. Although the two passages are created with identical musical content, a Lilypond<sup>9</sup> stylesheet changes the graphic result, removing beams, dots, rests, stems, and tuplet brackets. Abjad does not restrict the end result of a composer's score, but the choice of how the music is represented is an important one. It is fundamental to certain composers who use Abjad that the user may systematically choose when and how notation changes throughout the course of a score to better represent the desired influence of the visual document upon the interpreter. The deeply personal nature of a composer's chosen notation leads me to doubt that a dedicated user of Abjad would be satisfied with an edition of their score that was re-engraved by a copyist or was forced to match the house style of a publisher.<sup>10</sup>

In the past, a composer may have considered manipulating parameters of sound as a type of algebra of musical generation. Later, the Parisian Spectral composers took a more holistic view of musical parameters, taking a scientific view of sound as their model for musical composition. With the Abjad ideology of FSC, any and all graphic elements that appear on the page of a score may be treated as manipulable compositional entities. This means that transformational and developmental processes within a given piece might only occur in a parameter of performance mode rather than in more traditional musical materials such as pitch and rhythm. In this way, I am not composing with parameters of sound, rather I am composing with parameters of performance practices. Sound is no longer my model for music; bodily action takes its place. As an example, consider the following passage from my composition *Cthar* for two cellos.

---

<sup>9</sup>[lilypond.org](http://lilypond.org)

<sup>10</sup>This is not to say that compromises are impossible, but to highlight that the personality of the composer is transparent within the notation of a score.



Not only are the traditional parameters of articulation, intensity, pitch, and rhythm being manipulated programmatically, but so are features such as bow speed, bowing technique, and string contact point.<sup>11</sup>

### 1.2.2 Composition as Software Development

The second principle that is shared by many composers who use Abjad is Composition as Software Development. This applies to the kinds of workflows Abjad makes possible. While composers may use Abjad in many different ways, with Composition as Software Development the composer is no longer a simple user of a software, but a software developer in their own right.

Each composition is developed in the way a programmer writes software. Various reservoirs of material and processes can be defined and called at specific times throughout the composition process to be assembled into a final score. The composer also should not rely only on the built-in tools of Abjad for their music, but should also write their own software to produce notation in very personal ways. Composers should also do their best to follow established best practices of Python development for the sake of the clarity of the source code and for the possibility of expanding upon previously written code.<sup>12</sup> An added benefit of this methodology is that it is common for software developers to house their code in online repositories such as GitHub. These repositories allow for the perusal not only of others' source code, but also for the viewing of a history of the changes to

---

<sup>11</sup>While this piece uses a pseudo-tablature notation, it is not the only method by which non-traditional graphic elements can be manipulated in Abjad, although several composers who use Abjad have had significant periods where tablature was the solution to the notation of various parametric processes.

<sup>12</sup>For instance, composers should do their best to encapsulate functions that can be reused, rather than writing the same loop many times.

each file as a project develops.<sup>13</sup>

A final aspect of this principle addresses the issue of the user interface. If composing music with Abjad is akin to software design, then the resulting score is the Graphic User Interface<sup>14</sup> of the program. The user is able to interact with the software through their interpretation of the score. Just as a professional software developer would take great care to ensure that the GUI of a program is pristine, so do many Abjad composers. My reliance upon, and even integration of, technology in my compositional process and aesthetic could be linked to the use of the computer by spectral composers for models of sound or the use of computers by Iannis Xenakis and Brian Ferneyhough for its ability to help process abstractions. In contrast with these historical uses of the computer for music composition, the speed of modern computation allows for the rapid prototyping of material, allowing the composer to compose iteratively, one step at a time, instead of composing the entire process of a composition in one sitting.

### 1.3 Genesis of *GUERRERO*

Although *GUERRERO* was neither my first completed composition with Abjad nor the first I conceptualized, it began as a folder of some of the earliest experiments that I used to explore my own idiosyncratic workflow with Abjad. At the time, I had been studying the music of Francisco Guerrero Marín, a Spanish composer whose music is not well known in the United States. Specifically, I was studying his work Rhea[6] for twelve saxophones. Guerrero's music tended to be post-serial in construction, relying on many serial procedures for the organization of harmonic material with the added concern of combinatorial arrangements of many parameters of sound. While his early works are primarily concerned with combinatorial manipulation, his later work is often based on musical mappings of fractal models. While I have found no complete analyses of his works in English, my study of Rhea leads me to believe that it is both a combinatorial work and an early fractal work. The seven-term axiom[2] that appears in many of his pieces could be considered a kind of

---

<sup>13</sup>Because of this, a full history of my works composed in Abjad can be found in my repositories at [github.com/GregoryREvans](https://github.com/GregoryREvans). This is intended to keep with Abjad's open-source philosophy and for pedagogical purposes.

<sup>14</sup>GUI

self-similar process. Because I had the power of the Python programming language at my disposal for musical purposes, I decided to experiment with combinatorics, fractal models, and cycles as an homage to Guerrero in my own piece for twelve saxophones.

Before I began serious work on the piece, I asked Joey Speranzo, the conductor of the Frost Saxophone Ensemble, if there would be any possibility for a reading session of the piece once I completed it. We later agreed instead to premiere the piece on one of the ensemble's concerts. During the composition process, the number of performers in the ensemble steadily grew from twelve to twenty-one. Since much of my combinatorial planning relied heavily on the number twelve, the manipulation through combinatorics was abandoned as a primary feature in this piece, but fractals and cycles remained. Certainly, other numbers could have replaced twelves as the number of significance, but I chose to forego this option because of the gradual change of the ensemble size.<sup>15</sup>

## 1.4 Initial Concepts and Materials

*GUERRERO* also has an extra-musical inspiration. I had been reading a number of books on astronomy concurrently with my work on the piece, and I was interested in the physics of light. I was interested in writing a piece of music that had a constant, such as light, that could be warped, absorbed, perturbed, reflected, and refracted. This established the foundation of pitch material in this work.

At first, I experimented with using a pure harmonic spectrum as the fundamental timbre of this piece. I also experimented with harmonies derived from ring modulation.<sup>16</sup> In the end, I chose a harmony formed from a twelve-tone row of fixed register by voicing it across the entire ensemble, because I was interested in an auditory effect that felt less weighted to a tonic or fundamental, but felt more evenly distributed in density. This chord no longer functioned as a tone-row but as

---

<sup>15</sup>The original twelve member ensemble initially grew to fourteen, followed by sixteen, nineteen, and finally twenty-one.

<sup>16</sup>My interest in these spectral techniques at the time likely stemmed from my recent exposure to OM. At this stage, I was still trying to use Abjad as if it were a “better” version of OM.

a harmony-timbre from which all other material is in some way derived. The chord in question is Link Chord #23. This particular chord also features elements of parallel inversion around the axis of the central perfect fifth. The interval at the bottom of the chord is a minor second while the interval just above the central axis is a major seventh. The second interval from the bottom is a major third and the second interval above the central axis is a minor sixth. Finally, the interval just below the central axis is a major second and the topmost interval is a minor seventh. It was my intention that this element of parallel inversion would allow for a greater relationship between materials in different registers of the ensemble because of the similarity in harmonic content, while keeping the kind of smooth, unweighted harmony based on my previous criteria.

The Link chords are a set of all-interval, twelve-note chords with a contiguous statement of the all-trichord hexachord<sup>17</sup> somewhere within it.[4] It was originally calculated by music theorist John F. Link. Initially, I intended to treat the all-trichord hexachord within my sonority combinatorially, but as stated previously, this was abandoned due to the expansion of the ensemble. The following image is the ATH in the ascending reduced form of (012478) beginning on middle C.

The total combinations of three-note groups reveal the presence of all Forte trichords.[5]

Because these trichords are derived from the ATH, each can be paired with a complementary trichord to complete the full ATH.

<sup>17</sup>ATH

## 1.5 Formal Structures in *GUERRERO*

In line with my ideas of how musical material could behave like light, I wanted to explore auditory mirages. I wanted to move smoothly between a number of musical textures in a progression I found satisfying, but in a way that was also subtle. I started with the global form of the piece, constructing a pattern of segments and selecting what kind of material should develop within them. After this, I organized a more local pattern of timespans<sup>18</sup> during which the material would be performed. Each of these timespans were tagged with the name of a music generator I had written and Abjad populated these timespans with material created by my generators.

I chose to gradually move from a very ambiguous state to something with more clarity and back toward ambiguity, but not all the way back to the original state. The point of greatest ambiguity in *GUERRERO* is at the beginning. The orchestrational timespan pattern of this passage is meant to reference a Shepard tone,[7] an auditory illusion of infinite motion. The piece opens with an “Invocation” featuring an ascending quarter-tone scale performed as a glissando. The octave is never reached because the scale begins anew once it arrives one quarter step away from the octave. No satisfying harmonic consistency is established and the orchestration potentially gives the illusion of the Shepard tone’s endless motion.



---

<sup>18</sup>See chapter two for diagrams of these timespans.

The cascading rhythms of the Invocation are created from two taleas, which are cyclic rhythm patterns that develop independently of harmonic material that originated in music of the middle ages.[3]

$\text{♩} = 60$

Invocation

Soprano  
sopr.  
sopr.

$\text{mp}$        $\text{ff}$        $\text{5:4}$        $\text{7:8}$   
 $\text{mp}$        $\text{ff}$        $\text{5:4}$        $\text{7:8}$   
 $\text{mf}$        $f$        $\text{5:6}$        $f$   
 $\text{mf}$        $f$        $\text{5:4}$        $\text{ff}$        $\text{mf}$

Harmonically, the quarter-tone scale is applied to all notes in both taleas.<sup>19</sup> The cyclic rhythmic patterns of the first and second talea are  $\frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{5}{16}, \frac{3}{16}, \frac{2}{16}, \frac{4}{16}$ ,<sup>20</sup> and  $\frac{4}{8}, \frac{3}{8}, -\frac{1}{8}, \frac{2}{8}$  respectively. Negative fractions represent a duration that is always written as a rest. Another important concept taken from ancient music is that of prolation. The term prolation is taken from the Latin *Prolatio* coined by Philippe de Vitry.[3] With the coming of the Ars Nova, came new resources in music notation. For the first time, there was very little ambiguity in how rhythms should be performed. With this new notation came new terminology to describe the surface of the new music. *Modus* refers to the division of the long, *tempus* refers to the division of the breve, and the division of the semibreve was called *prolatio*. At this point, this was the lowest level of division possible. With perfect or imperfect divisions, these layers of time could be divided into groups of three or groups of two. Now, the modern, more general use of the term *prolation* refers to the depth of division of temporal units of music. Tuplets with fractions of the same ratio are said to have the same prolation, thus the duration of a given rhythm within a tuplet of 3:2 is temporally equivalent the duration of the same rhythm written inside of a bracket of 9:6. Unlike the Ars Nova, there is no restriction to divisions of three and two in modern music. At will, composers may use divisions of

<sup>19</sup>The plural form of talea is actually “taleae,” but it is the convention in Abjad to use the more convenient, anglicized plural “taleas.”

<sup>20</sup>Where  $\frac{1}{16}$  refers to one sixteenth note and  $\frac{1}{8}$  refers to one eighth note etc.

any number they choose. The following is an example talea of  $\frac{1}{16}$ ,  $\frac{2}{16}$ ,  $\frac{3}{16}$ , and  $\frac{4}{16}$ .<sup>21</sup>



A cycle of tuplet brackets is applied to each timespan containing rhythmic material. Portions of the defined durations are shifted over to fit inside of the newly prolated container.



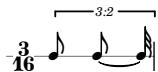
The brackets are then respelled in a way that redundant tuplets are removed.



The fraction of the bracket is calculated based on the duration of the timespan, but these fractions are written as pure prolations. This means that a tuplet that could be written as 9:6 is always written in a reduced form of 3:2. Thus the following passage



would be rewritten as this



instead. The tuplet bracket cycle applied to the first talea in the Invocation passage of *GUERRERO* is 0, 1, -1, where 0 represents no change in prolation, 1 represents an augmentation, and -1 represents a diminution. The tuplet bracket cycle applied to the second talea is -1, 0, -1, 1, 0.

Each talea is assigned unique dynamic trajectories. The first trajectory is a sudden swell from mezzo piano to fortissimo and the second trajectory is a smooth crescendo from mezzo forte to forte. Example. Originally, every note that is part of the first talea was given a tenuto marking. The second talea's cycle is one tenuto followed by four notes without articulations. In the final version of the piece, only the second talea's articulation cycle is present.

---

<sup>21</sup>The Abjad tools used to produce the following three examples are Trevor Bača's *rmakers*. They are described in great detail on the Abjad website as well as in Josiah Oberholtzer's dissertation *A Computational Model of Music Composition*.

This kind of process is characteristic of each section of *GUERRERO*: a rhythmic process is grouped with a pitch process, dynamic trajectory, and articulation pattern. These units alternate and interrupt each other. Each section was composed in isolation,<sup>22</sup> so these processes remain restricted to their respective segment.<sup>23</sup> As the piece progresses, some segments appear that act as a recapitulation of previous material in a significantly varied form. For instance, section H reiterates the Shepard tone theme, this time with slower rhythms and a significant element of noise.

## 1.6 Processes in the Remaining Sections of *GUERRERO*

In section A, the fundamental harmony of Link-Chord #23 begins to appear, albeit in a highly distorted form. I had originally planned to emphasize a specific note in each of the twenty-one instruments. While this is not present in the final version of the piece, there remains a trace of that original design. When the fundamental harmony is voiced across the ensemble, those instruments that are assigned a note from the all-trichord hexachord are also assigned multiphonics. These multiphonics and their fingerings were taken from the Bärenreiter book *Die Spieltechnik des Saxophons*.<sup>[8]</sup> I originally approximated the sounding pitches of the multiphonics to quarter-tones, but eventually I settled on an eighth-tone approximation. These specific multiphonics were chosen because they contained literal pitches from the fundamental harmony and therefore usually contained members of the all-trichord hexachord within Link chord #23, or pitches that were close

<sup>22</sup>Evidence of this can be seen in the timespan patterns in chapter two.

<sup>23</sup>In Abjad lingo, the word segment refers to a passage of music that was composed in isolation from any other passage. Though not required, these segments often take shape as separate files that are stitched together in Lilypond to produce the final score.

approximations. Exact pitches from Link chord #23 are highlighted in green, while approximate pitches are highlighted in blue. Some of these multiphonics also contain extraneous pitches and they are notated in black.

The musical score shows a sequence of multiphonics for five voices: soprano, alto, tenor, baritone, and bass. Each voice has three entries. The first entry for soprano is at dynamic ff, the second at f, and the third at mf. The alto entries are at ff, f, and mf. The tenor entries are at ff, f, and mf. The baritone entries are at ff, f, and mf. The bass entry is at ff. The bass voice ends with a label 'lowA'.

The rhythmic material for this passage began as a sustained note for the duration of each timespan, but evolved into taleas of long durations in order to keep the rhythmic landscape more amorphous. The passage gradually increases in rhythmic density while clarifying the harmony by reducing the number of multiphonics played.

In Section B, the harmony has almost settled into a recognizable form. The pitches are derived from a musical mapping of Brownian Motion, with each step scaled to a quarter-tone. Brownian Motion, sometimes graphed in the form of a Random Walk, is a description of a chaotic system where particles move in seemingly unpredictable ways. However, when statistically analyzed, this chaotic system will form a normal distribution, one of the most common distributions found in nature. Another characteristic of Brownian Motion is that if any slice of a random walk is analyzed, it still forms a normal distribution. This property often causes Brownian Motion to be considered a fractal model. The shape of these walks is governed by the fact that a physical system is presupposed. Particles cannot teleport from one location to the next. They must first pass through spaces adjacent to their immediate position. In *GUERRERO*, these random walk mappings begin and center around chord tones, but are erratic so that the harmony is still unclear.

The musical score for Section B consists of three staves, all labeled 'Tenor 1'. The top staff starts with a dynamic marking of  $\text{mf}$ . The middle staff starts with  $\text{pp}$ . The bottom staff starts with  $\text{pp}$ . All staves feature eighth-note patterns with various dynamics and performance instructions such as ' $< >$ ' and ' $> <$ ' placed above the notes.



Section C acts a quick transition between the texture of sections B and D. In this Section, I increase the density of material and introduce the air tones that become prominent later in the piece. Section D is emphasized by a focus on a continuum of pitch and noise. While every pitch in this section is drawn from Link chord #23, the clarity of its statement is hindered and absorbed by the surrounding noise. The tempo is much faster in this passage, creating a cloud of sound.

Section E has the most prominent statement of the fundamental harmony. Brownian models are mapped not to steps in a scale, but to pitches in the fundamental chord. Some dyads are also isolated as trills. This is the most rhythmically dense passage of the piece, but the harmony is clearest. This Brownian process and trill isolation is continued through Section F although the texture is much thinner. The non-chord tones are derived from the multiphonics and threaten to overcome the stability of the fundamental harmony.

Section G features a return to random walks beginning and centering around chord tones, as found earlier in section C. This time, the glissandi are scaled to half-tones rather than quarter-tones. This is intended to emphasize the chord tones by placing them in the context of a twelve-note chromatic space rather than the distorted twenty-four note chromatic scale presented earlier.

Section I is slow and returns to only mapping pitches in the fundamental chord, while the concluding section, Section J, returns to the texture of Section C. This initially transitional character has been transformed from a bridge into a climax, as the trajectory of the piece is suddenly halted. While there is not an intended teleological trajectory within this piece, the motion from ambiguity to clarity with a return to ambiguity feels extremely satisfying to me.

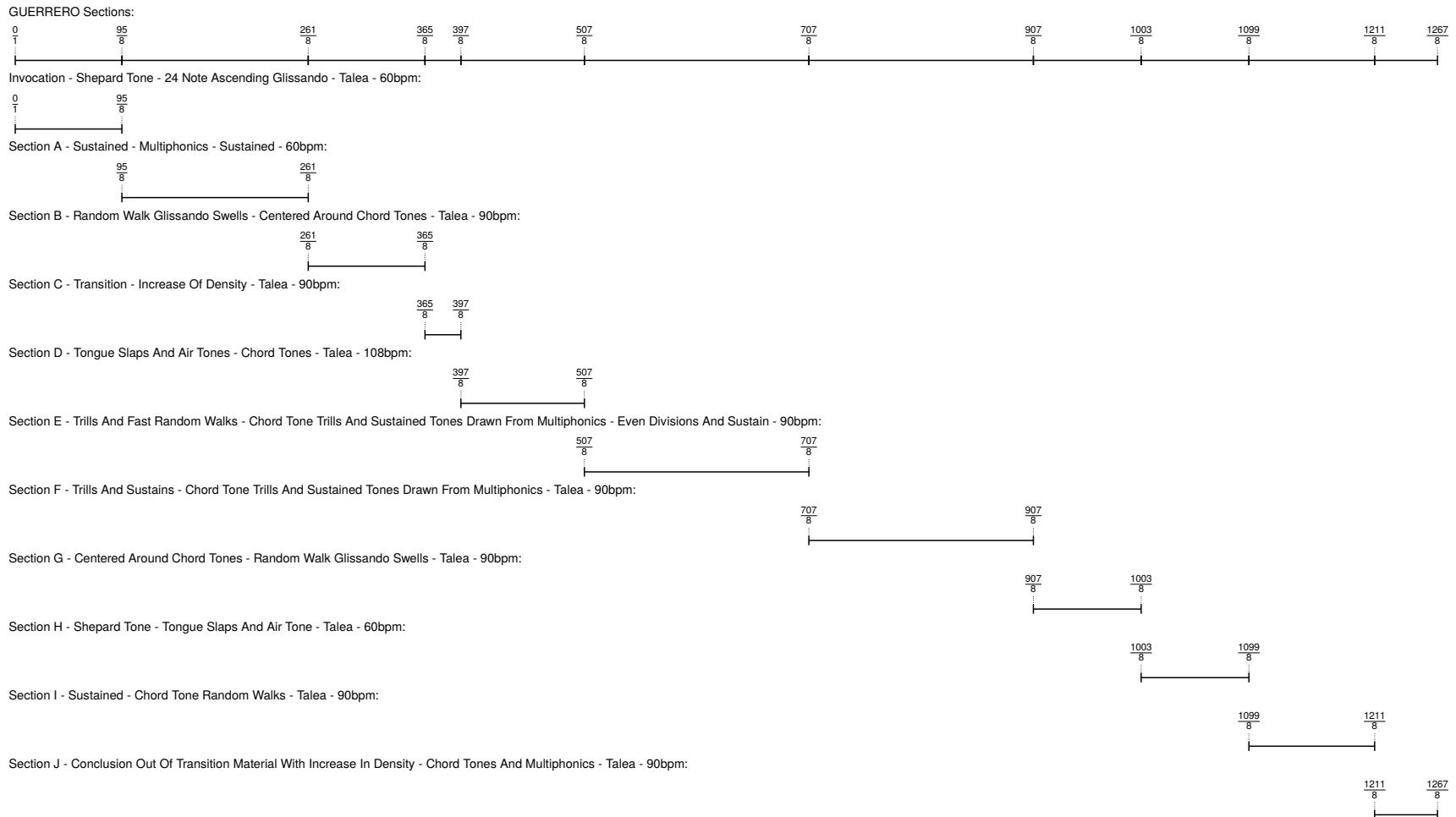
## 1.7 Future Implications for My Work

After composing *GUERRERO*, many of my methodologies for composing music with Abjad have changed. Now, I take greater care with the distribution of materials. To do this I am experimenting with the use of Markovian randomness. I have streamlined the process of timespan generation, allowing me to spend more time on other decision-making processes and overall, I have streamlined the file system for my compositions. With these new workflows and new tools for rhythm and pitch generation, I am excited to expand on my computational music in the future.

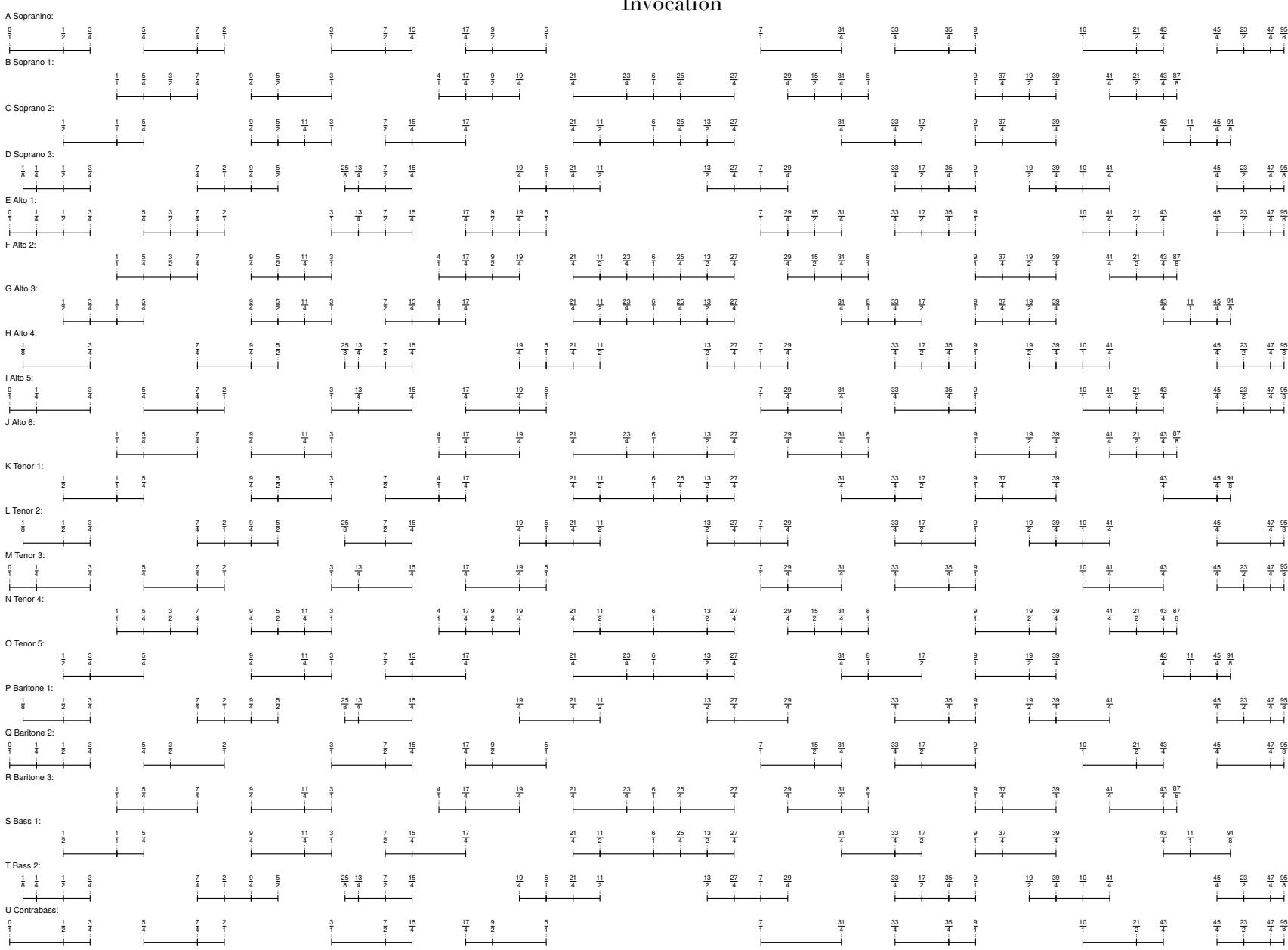
## **Chapter 2**

# **Supplemental Illustrations**

### **2.1 Global Form**

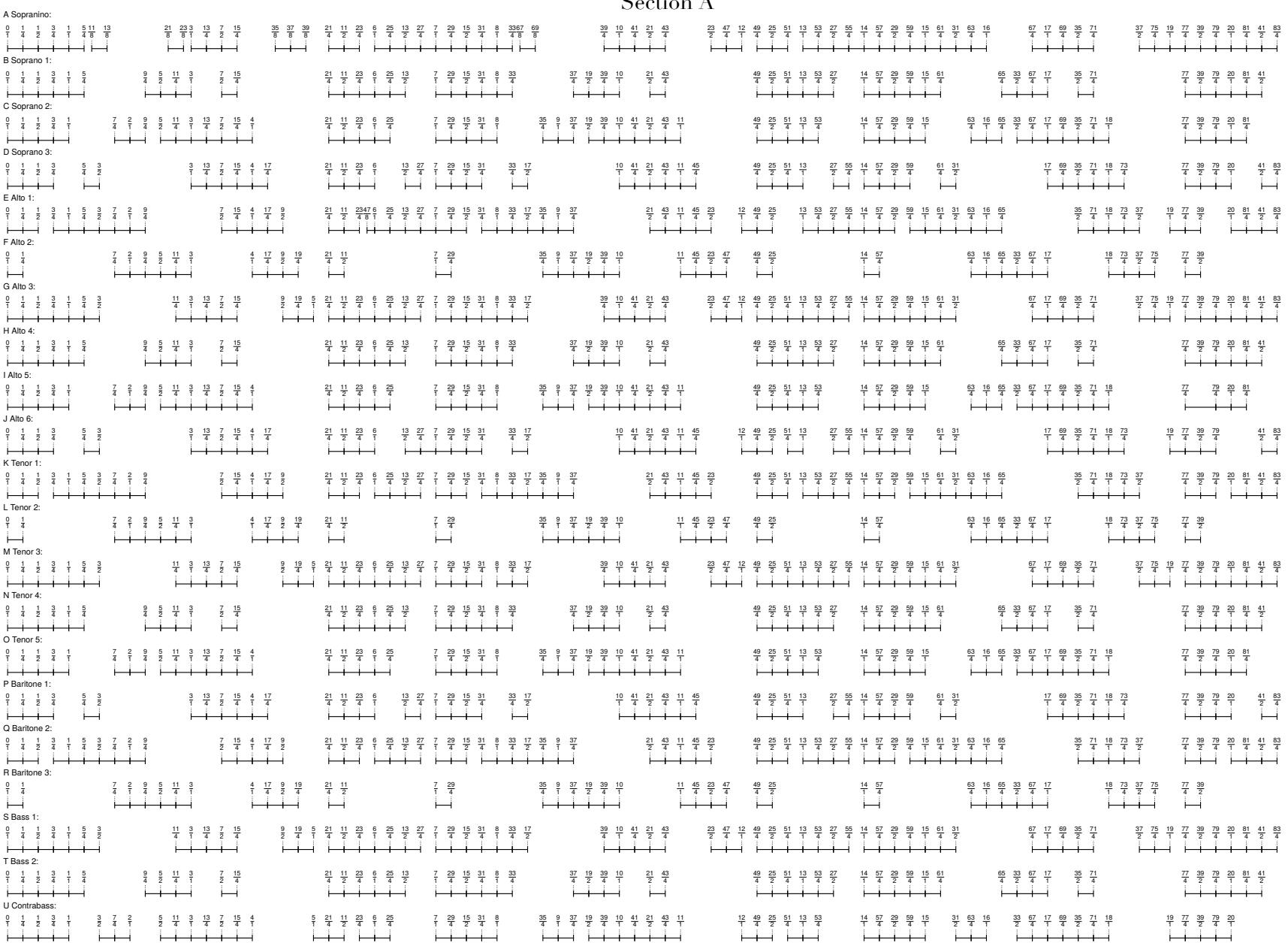


## 2.2 Invocation

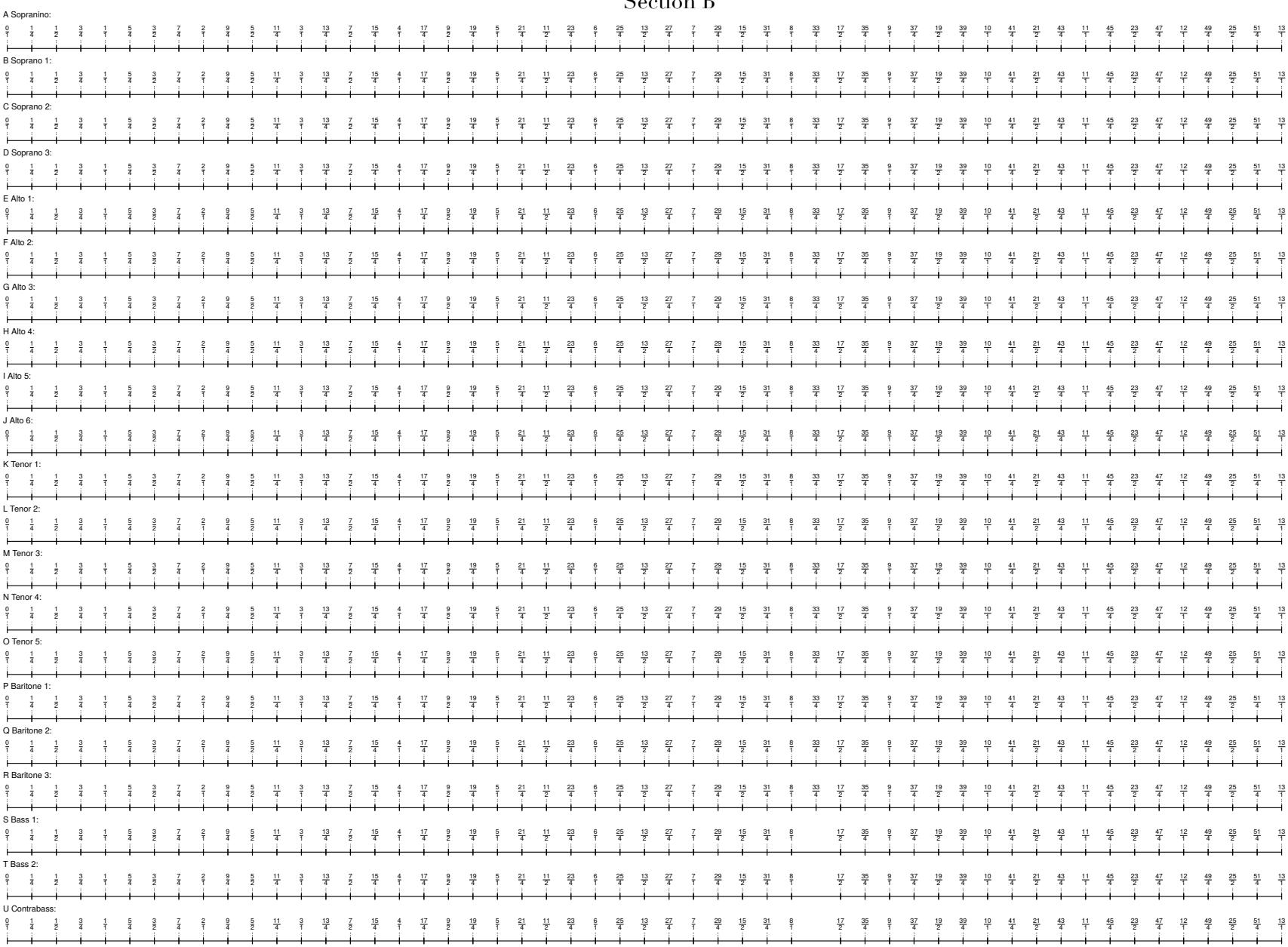


## 2.3 Section A

## Section A

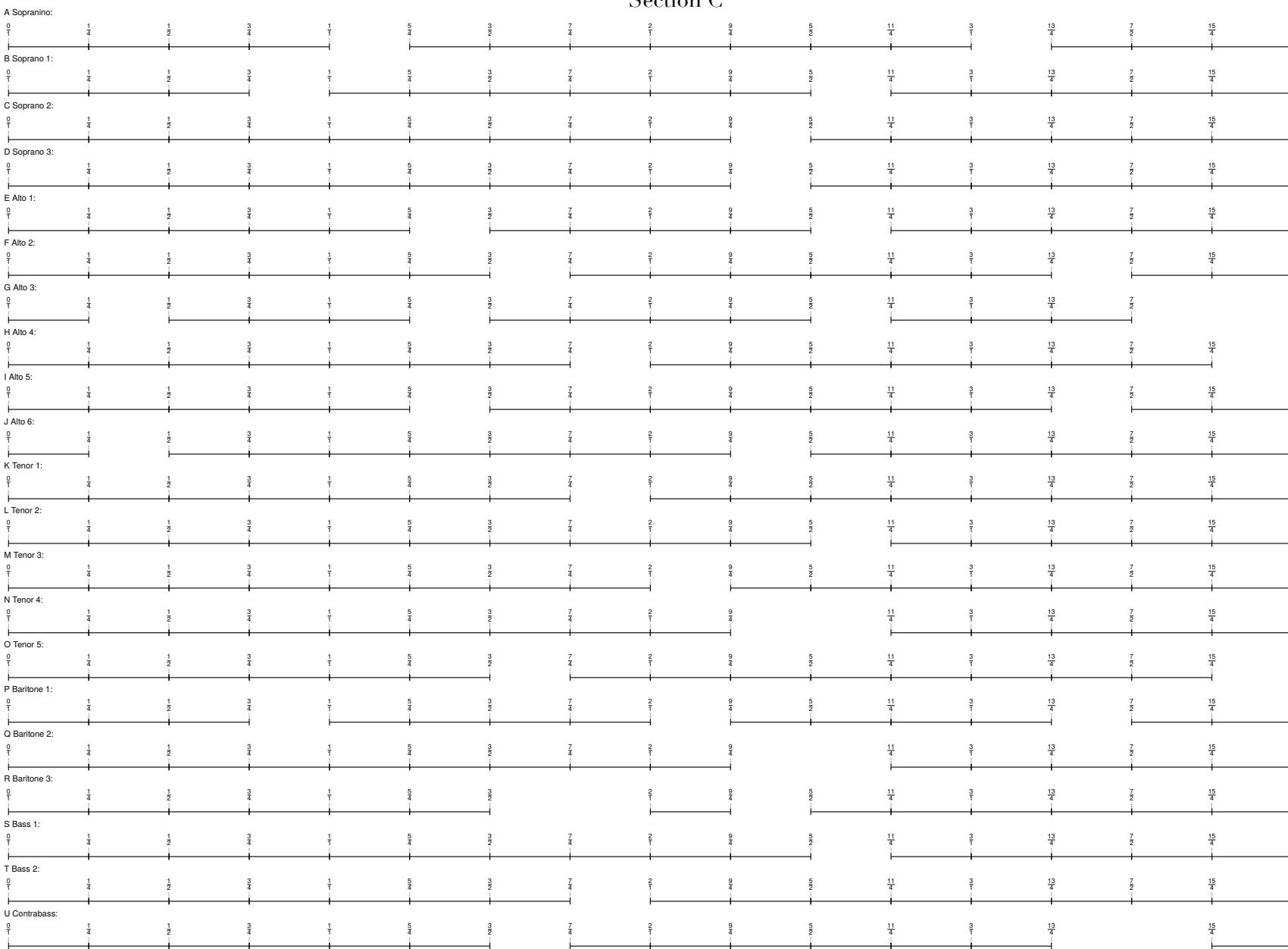


## 2.4 Section B



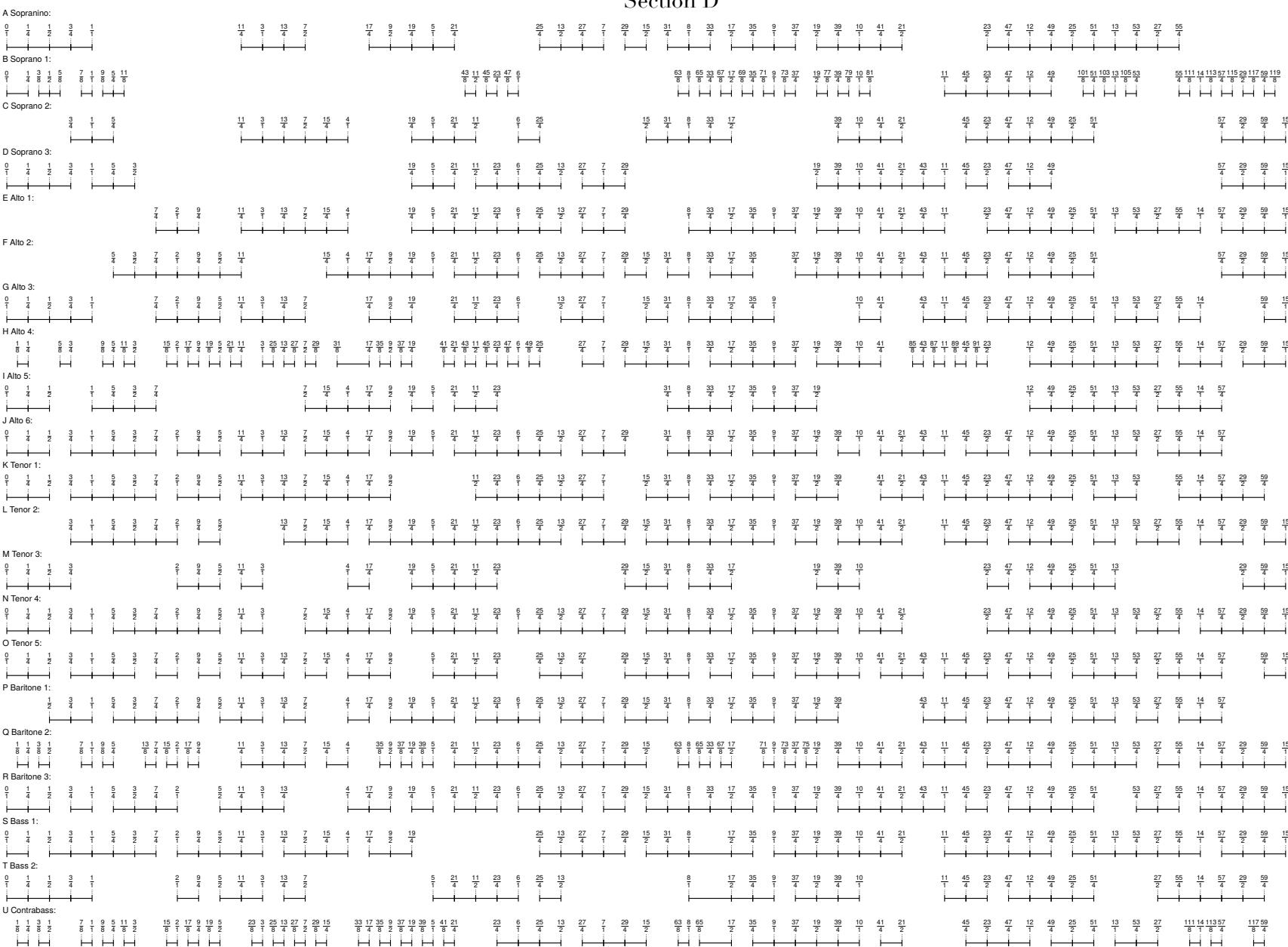
## 2.5 Section C

## Section C



## 2.6 Section D

## Section D



## 2.7 Section E



## 2.8 Section F

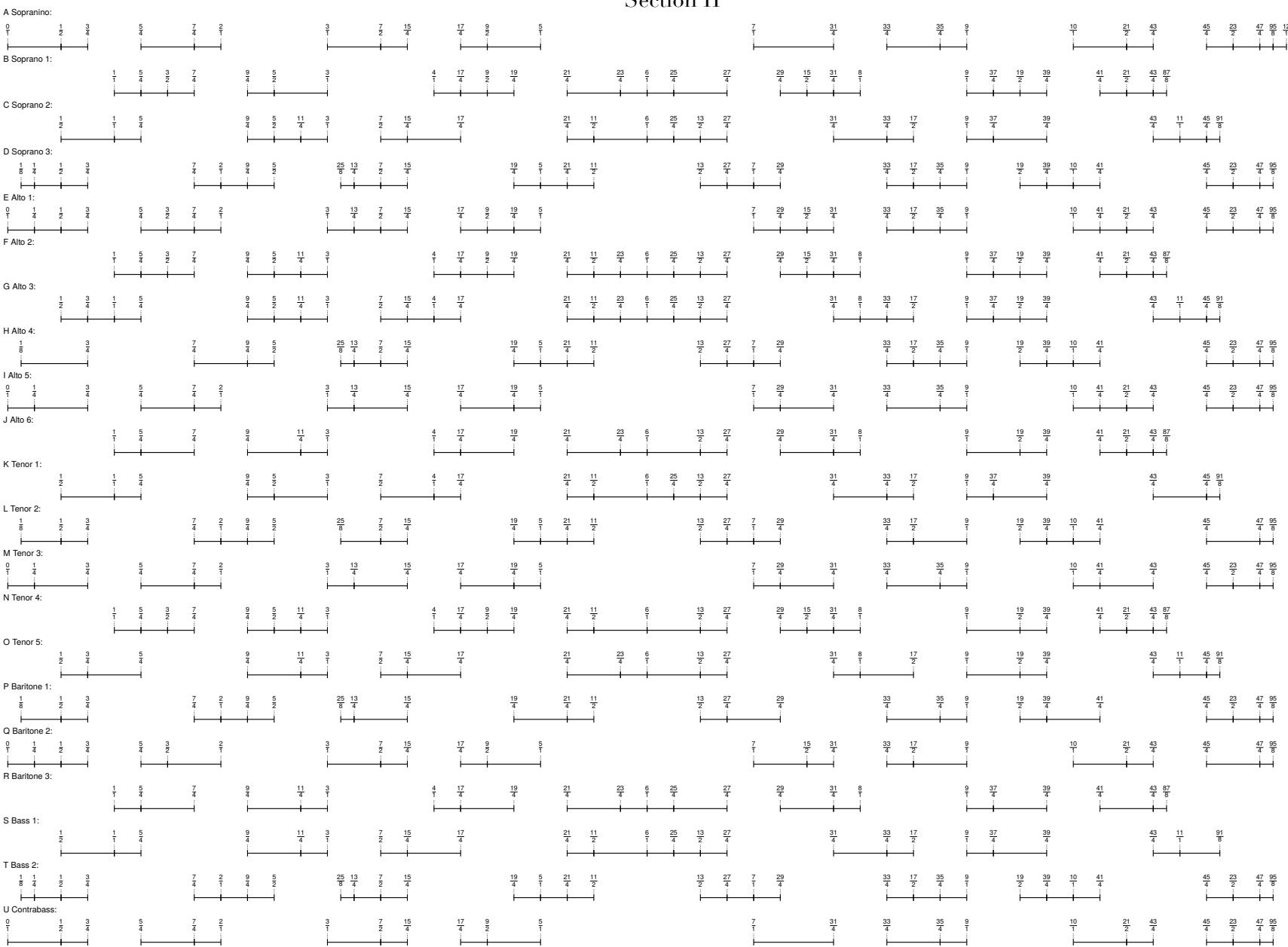


## 2.9 Section G

## Section G

## 2.10 Section H

## Section H



## 2.11 Section I



## 2.12 Section J

## Section J



# Chapter 3

## Source Code

### 3.1 Invocation

```
1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11
12 print('Interpreting file ...')
13
14 time_signatures = [
15     abjad.TimeSignature(pair) for pair in [
16         (5, 4), (4, 4), (3, 4), (5, 4), (4, 4), (3, 4),
17         (3, 4), (4, 4), (5, 4), (3, 4), (4, 4), (9, 8),
18     ]
19 ]
20
21 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
22
23 def cyc(lst):
24     count = 0
25     while True:
26         yield lst[count%len(lst)]
27         count += 1
28
29 def grouper(lst1, lst2):
30     def cyc(lst):
31         c = 0
32         while True:
33             yield lst[c%len(lst)]
34             c += 1
35     lst1 = cyc(lst1)
36     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2]
```

```

]
37
38 soprano_notes = [14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20,
                   20.5, 21, 21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25, 25.5]
39 soprano_1_notes = [15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21,
                     21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25, 25.5, 26, 26.5]
40 soprano_2_notes = [11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17,
                     17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5, 23]
41 soprano_3_notes = [11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17,
                     17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5]
42 alto_1_notes = [14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20,
                  20.5, 21, 21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25, 25.5]
43 alto_2_notes = [10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16,
                  16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5]
44 alto_3_notes = [6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13,
                  13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5]
45 alto_4_notes = [1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9,
                  9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13]
46 alto_5_notes = [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9,
                  9.5, 10, 10.5, 11, 11.5, 12, 12.5]
47 alto_6_notes = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8,
                  8.5, 9, 9.5, 10, 10.5, 11, 11.5]
48 tenor_1_notes = [7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5,
                   14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5]
49 tenor_2_notes = [3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5,
                   11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5]
50 tenor_3_notes = [-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,
                   7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5]
51 tenor_4_notes = [0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8,
                   8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12]
52 tenor_5_notes = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5,
                   8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5]
53 baritone_1_notes = [2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9,
                   9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5]
54 baritone_2_notes = [-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5,
                   5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5]
55 baritone_3_notes = [-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5,
                   5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10]
56 bass_1_notes = [-0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7,
                   7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11]
57 bass_2_notes = [-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,
                   7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5]
58 contrabass_notes = [6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5,
                   13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5]
59
60 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
61     talea=abjadext.rmakers.Talea(
62         counts=[1, 1, 1, 5, 3, 2, 4],
63         denominator=16,
64     ),
65     beamSpecifier=abjadext.rmakers.BeamSpecifier(
66         beamDivisionsTogether=True,
67         beamRests=False,
68     ),
69     extraCountsPerDivision=[0, 1, -1, ],
70     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
71         trivialize=True,
72         extractTrivial=True,
73         rewriteRestFilled=True,

```

```

74     rewrite_sustained=True,
75   ),
76 )
77
78 rmaker_two = abjadext.rmakers.TaleaRhythmMaker(
79   talea=abjadext.rmakers.Talea(
80     counts=[4, 3, -1, 2],
81     denominator=8,
82   ),
83   beamSpecifier=abjadext.rmakers.BeamSpecifier(
84     beamDivisionsTogether=True,
85     beamRests=False,
86   ),
87   extraCountsPerDivision=[-1, 0, -1, 1, 0, ],
88   tupletSpecifier=abjadext.rmakers.TupletSpecifier(
89     trivialize=True,
90     extractTrivial=True,
91     rewriteRestFilled=True,
92     rewriteSustained=True,
93   ),
94 )
95
96 attachmentHandlerOne = AttachmentHandler(
97   startingDynamic='mp',
98   endingDynamic='ff',
99   hairpin='<|',
100 )
101
102 attachmentHandlerTwo = AttachmentHandler(
103   startingDynamic='mf',
104   endingDynamic='f',
105   hairpin='<',
106   articulationList=['tenuto', ' ', ' ', ' ', ' ', ' ', ],
107 )
108
109 # Initialize MusicMakers with the rhythm-makers.
110 #####sopranino#####
111 sopranino_musicmakerOne = MusicMaker(
112   rmaker=rmakerOne,
113   pitches=sopraninoNotes,
114   continuous=True,
115   attachmentHandler=attachmentHandlerOne,
116 )
117 sopranino_musicmakerTwo = MusicMaker(
118   rmaker=rmakerTwo,
119   pitches=sopraninoNotes,
120   continuous=True,
121   attachmentHandler=attachmentHandlerTwo,
122 )
123 #####soprano_one#####
124 sopranoOne_musicmakerOne = MusicMaker(
125   rmaker=rmakerOne,
126   pitches=soprano_1Notes,
127   continuous=True,
128   attachmentHandler=attachmentHandlerOne,
129 )
130 sopranoOne_musicmakerTwo = MusicMaker(
131   rmaker=rmakerTwo,
132   pitches=soprano_1Notes,

```

```
133     continuous=True,
134     attachment_handler=attachment_handler_two,
135 )
136 #####soprano_two#####
137 soprano_two_musicmaker_one = MusicMaker(
138     rmaker=rmaker_one,
139     pitches=soprano_2_notes,
140     continuous=True,
141     attachment_handler=attachment_handler_one,
142 )
143 soprano_two_musicmaker_two = MusicMaker(
144     rmaker=rmaker_two,
145     pitches=soprano_2_notes,
146     continuous=True,
147     attachment_handler=attachment_handler_two,
148 )
149 #####soprano_three#####
150 soprano_three_musicmaker_one = MusicMaker(
151     rmaker=rmaker_one,
152     pitches=soprano_3_notes,
153     continuous=True,
154     attachment_handler=attachment_handler_one,
155 )
156 soprano_three_musicmaker_two = MusicMaker(
157     rmaker=rmaker_two,
158     pitches=soprano_3_notes,
159     continuous=True,
160     attachment_handler=attachment_handler_two,
161 )
162 #####alto_one#####
163 alto_one_musicmaker_one = MusicMaker(
164     rmaker=rmaker_one,
165     pitches=alto_1_notes,
166     continuous=True,
167     attachment_handler=attachment_handler_one,
168 )
169 alto_one_musicmaker_two = MusicMaker(
170     rmaker=rmaker_two,
171     pitches=alto_1_notes,
172     continuous=True,
173     attachment_handler=attachment_handler_two,
174 )
175 #####alto_two#####
176 alto_two_musicmaker_one = MusicMaker(
177     rmaker=rmaker_one,
178     pitches=alto_2_notes,
179     continuous=True,
180     attachment_handler=attachment_handler_one,
181 )
182 alto_two_musicmaker_two = MusicMaker(
183     rmaker=rmaker_two,
184     pitches=alto_2_notes,
185     continuous=True,
186     attachment_handler=attachment_handler_two,
187 )
188 #####alto_three#####
189 alto_three_musicmaker_one = MusicMaker(
190     rmaker=rmaker_one,
191     pitches=alto_3_notes,
```

```
192     continuous=True,
193     attachment_handler=attachment_handler_one,
194 )
195 alto_three_musicmaker_two = MusicMaker(
196     rmaker=rmaker_two,
197     pitches=alto_3_notes,
198     continuous=True,
199     attachment_handler=attachment_handler_two,
200 )
201 #####alto_four#####
202 alto_four_musicmaker_one = MusicMaker(
203     rmaker=rmaker_one,
204     pitches=alto_4_notes,
205     continuous=True,
206     attachment_handler=attachment_handler_one,
207 )
208 alto_four_musicmaker_two = MusicMaker(
209     rmaker=rmaker_two,
210     pitches=alto_4_notes,
211     continuous=True,
212     attachment_handler=attachment_handler_two,
213 )
214 #####alto_five#####
215 alto_five_musicmaker_one = MusicMaker(
216     rmaker=rmaker_one,
217     pitches=alto_5_notes,
218     continuous=True,
219     attachment_handler=attachment_handler_one,
220 )
221 alto_five_musicmaker_two = MusicMaker(
222     rmaker=rmaker_two,
223     pitches=alto_5_notes,
224     continuous=True,
225     attachment_handler=attachment_handler_two,
226 )
227 #####alto_six#####
228 alto_six_musicmaker_one = MusicMaker(
229     rmaker=rmaker_one,
230     pitches=alto_6_notes,
231     continuous=True,
232     attachment_handler=attachment_handler_one,
233 )
234 alto_six_musicmaker_two = MusicMaker(
235     rmaker=rmaker_two,
236     pitches=alto_6_notes,
237     continuous=True,
238     attachment_handler=attachment_handler_two,
239 )
240 #####tenor_one#####
241 tenor_one_musicmaker_one = MusicMaker(
242     rmaker=rmaker_one,
243     pitches=tenor_1_notes,
244     continuous=True,
245     attachment_handler=attachment_handler_one,
246 )
247 tenor_one_musicmaker_two = MusicMaker(
248     rmaker=rmaker_two,
249     pitches=tenor_1_notes,
250     continuous=True,
```

```
251     attachment_handler=attachment_handler_two ,  
252 )  
253 #####tenor_two#####  
254 tenor_two_musicmaker_one = MusicMaker(  
255     rmaker=rmaker_one ,  
256     pitches=tenor_2_notes ,  
257     continuous=True ,  
258     attachment_handler=attachment_handler_one ,  
259 )  
260 tenor_two_musicmaker_two = MusicMaker(  
261     rmaker=rmaker_two ,  
262     pitches=tenor_2_notes ,  
263     continuous=True ,  
264     attachment_handler=attachment_handler_two ,  
265 )  
266 #####tenor_three#####  
267 tenor_three_musicmaker_one = MusicMaker(  
268     rmaker=rmaker_one ,  
269     pitches=tenor_3_notes ,  
270     continuous=True ,  
271     attachment_handler=attachment_handler_one ,  
272 )  
273 tenor_three_musicmaker_two = MusicMaker(  
274     rmaker=rmaker_two ,  
275     pitches=tenor_3_notes ,  
276     continuous=True ,  
277     attachment_handler=attachment_handler_two ,  
278 )  
279 #####tenor_four#####  
280 tenor_four_musicmaker_one = MusicMaker(  
281     rmaker=rmaker_one ,  
282     pitches=tenor_4_notes ,  
283     continuous=True ,  
284     attachment_handler=attachment_handler_one ,  
285 )  
286 tenor_four_musicmaker_two = MusicMaker(  
287     rmaker=rmaker_two ,  
288     pitches=tenor_4_notes ,  
289     continuous=True ,  
290     attachment_handler=attachment_handler_two ,  
291 )  
292 #####tenor_five#####  
293 tenor_five_musicmaker_one = MusicMaker(  
294     rmaker=rmaker_one ,  
295     pitches=tenor_5_notes ,  
296     continuous=True ,  
297     attachment_handler=attachment_handler_one ,  
298 )  
299 tenor_five_musicmaker_two = MusicMaker(  
300     rmaker=rmaker_two ,  
301     pitches=tenor_5_notes ,  
302     continuous=True ,  
303     attachment_handler=attachment_handler_two ,  
304 )  
305 #####baritone_one#####  
306 baritone_one_musicmaker_one = MusicMaker(  
307     rmaker=rmaker_one ,  
308     pitches=baritone_1_notes ,  
309     continuous=True ,
```

```
310     attachment_handler=attachment_handler_one ,
311 )
312 baritone_one_musicmaker_two = MusicMaker(
313     rmaker=rmaker_two,
314     pitches=baritone_1_notes,
315     continuous=True,
316     attachment_handler=attachment_handler_two,
317 )
318 #####baritone_two#####
319 baritone_two_musicmaker_one = MusicMaker(
320     rmaker=rmaker_one,
321     pitches=baritone_2_notes,
322     continuous=True,
323     attachment_handler=attachment_handler_one,
324 )
325 baritone_two_musicmaker_two = MusicMaker(
326     rmaker=rmaker_two,
327     pitches=baritone_2_notes,
328     continuous=True,
329     attachment_handler=attachment_handler_two,
330 )
331 #####baritone_three#####
332 baritone_three_musicmaker_one = MusicMaker(
333     rmaker=rmaker_one,
334     pitches=baritone_3_notes,
335     continuous=True,
336     attachment_handler=attachment_handler_one,
337 )
338 baritone_three_musicmaker_two = MusicMaker(
339     rmaker=rmaker_two,
340     pitches=baritone_3_notes,
341     continuous=True,
342     attachment_handler=attachment_handler_two,
343 )
344 #####bass_one#####
345 bass_one_musicmaker_one = MusicMaker(
346     rmaker=rmaker_one,
347     pitches=bass_1_notes,
348     continuous=True,
349     attachment_handler=attachment_handler_one,
350 )
351 bass_one_musicmaker_two = MusicMaker(
352     rmaker=rmaker_two,
353     pitches=bass_1_notes,
354     continuous=True,
355     attachment_handler=attachment_handler_two,
356 )
357 #####bass_two#####
358 bass_two_musicmaker_one = MusicMaker(
359     rmaker=rmaker_one,
360     pitches=bass_2_notes,
361     continuous=True,
362     attachment_handler=attachment_handler_one,
363 )
364 bass_two_musicmaker_two = MusicMaker(
365     rmaker=rmaker_two,
366     pitches=bass_2_notes,
367     continuous=True,
368     attachment_handler=attachment_handler_two,
```

```

369 )
370 #####contrabass#####
371 contrabass_musicmaker_one = MusicMaker(
372     rmaker=rmaker_one,
373     pitches=contrabass_notes,
374     continuous=True,
375     attachment_handler=attachment_handler_one,
376 )
377 contrabass_musicmaker_two = MusicMaker(
378     rmaker=rmaker_two,
379     pitches=contrabass_notes,
380     continuous=True,
381     attachment_handler=attachment_handler_two,
382 )
383
384 silence_maker = abjadext.rmakers.NoteRhythmMaker(
385     division_masks=[
386         abjadext.rmakers.SilenceMask(
387             pattern=abjad.index([0], 1),
388             ),
389         ],
390     )
391
392 class MusicSpecifier:
393
394     def __init__(self, music_maker, voice_name):
395         self.music_maker = music_maker
396         self.voice_name = voice_name
397
398     print('Collecting timespans and rmakers ...')
399
400 voice_1_timespan_list = abjad.TimespanList([
401     abjad.AnnotatedTimespan(
402         start_offset=start_offset,
403         stop_offset=stop_offset,
404         annotation=MusicSpecifier(
405             music_maker=music_maker,
406             voice_name='Voice 1',
407             ),
408     )
409     for start_offset, stop_offset, music_maker in [
410         [(0, 4), (2, 4), sopranino_musicmaker_one],
411         [(2, 4), (3, 4), sopranino_musicmaker_one],
412         [(5, 4), (7, 4), sopranino_musicmaker_one],
413         [(7, 4), (8, 4), sopranino_musicmaker_one],
414         [(12, 4), (14, 4), sopranino_musicmaker_two],
415         [(14, 4), (15, 4), sopranino_musicmaker_two],
416         [(17, 4), (18, 4), sopranino_musicmaker_one],
417         [(18, 4), (20, 4), sopranino_musicmaker_one],
418         [(28, 4), (31, 4), sopranino_musicmaker_two],
419         [(33, 4), (35, 4), sopranino_musicmaker_two],
420         [(35, 4), (36, 4), sopranino_musicmaker_two],
421         [(40, 4), (42, 4), sopranino_musicmaker_one],
422         [(42, 4), (43, 4), sopranino_musicmaker_one],
423         [(45, 4), (46, 4), sopranino_musicmaker_two],
424         [(46, 4), (47, 4), sopranino_musicmaker_two],
425         [(47, 4), (95, 8), sopranino_musicmaker_two],
426         # [(95, 8), (96, 8), silence_maker],
427     ]

```

```

428 ])
429
430 voice_2_timespan_list = abjad.TimespanList([
431     abjad.AnnotatedTimespan(
432         start_offset=start_offset,
433         stop_offset=stop_offset,
434         annotation=MusicSpecifier(
435             music_maker=music_maker,
436             voice_name='Voice 2',
437         ),
438     )
439     for start_offset, stop_offset, music_maker in [
440         [(4, 4), (5, 4), soprano_one_musicmaker_two],
441         [(5, 4), (6, 4), soprano_one_musicmaker_two],
442         [(6, 4), (7, 4), soprano_one_musicmaker_two],
443         [(9, 4), (10, 4), soprano_one_musicmaker_one],
444         [(10, 4), (12, 4), soprano_one_musicmaker_one],
445         [(16, 4), (17, 4), soprano_one_musicmaker_two],
446         [(17, 4), (18, 4), soprano_one_musicmaker_two],
447         [(18, 4), (19, 4), soprano_one_musicmaker_two],
448         [(21, 4), (23, 4), soprano_one_musicmaker_one],
449         [(23, 4), (24, 4), soprano_one_musicmaker_one],
450         [(24, 4), (25, 4), soprano_one_musicmaker_one],
451         [(25, 4), (27, 4), soprano_one_musicmaker_one],
452         [(29, 4), (30, 4), soprano_one_musicmaker_two],
453         [(30, 4), (31, 4), soprano_one_musicmaker_two],
454         [(31, 4), (32, 4), soprano_one_musicmaker_two],
455         [(36, 4), (37, 4), soprano_one_musicmaker_one],
456         [(37, 4), (38, 4), soprano_one_musicmaker_one],
457         [(38, 4), (39, 4), soprano_one_musicmaker_one],
458         [(41, 4), (42, 4), soprano_one_musicmaker_two],
459         [(42, 4), (43, 4), soprano_one_musicmaker_two],
460         [(43, 4), (87, 8), soprano_one_musicmaker_two],
461     ]
462 ])
463
464 voice_3_timespan_list = abjad.TimespanList([
465     abjad.AnnotatedTimespan(
466         start_offset=start_offset,
467         stop_offset=stop_offset,
468         annotation=MusicSpecifier(
469             music_maker=music_maker,
470             voice_name='Voice 3',
471         ),
472     )
473     for start_offset, stop_offset, music_maker in [
474         [(2, 4), (4, 4), soprano_two_musicmaker_one],
475         [(4, 4), (5, 4), soprano_two_musicmaker_one],
476         [(9, 4), (10, 4), soprano_two_musicmaker_two],
477         [(10, 4), (11, 4), soprano_two_musicmaker_two],
478         [(11, 4), (12, 4), soprano_two_musicmaker_two],
479         [(14, 4), (15, 4), soprano_two_musicmaker_two],
480         [(15, 4), (17, 4), soprano_two_musicmaker_two],
481         [(21, 4), (22, 4), soprano_two_musicmaker_one],
482         [(22, 4), (24, 4), soprano_two_musicmaker_one],
483         [(24, 4), (25, 4), soprano_two_musicmaker_two],
484         [(25, 4), (26, 4), soprano_two_musicmaker_two],
485         [(26, 4), (27, 4), soprano_two_musicmaker_two],
486         [(31, 4), (33, 4), soprano_two_musicmaker_one],

```

```

487     [(33, 4), (34, 4), soprano_two_musicmaker_one],
488     [(36, 4), (37, 4), soprano_two_musicmaker_one],
489     [(37, 4), (39, 4), soprano_two_musicmaker_one],
490     [(43, 4), (44, 4), soprano_two_musicmaker_two],
491     [(44, 4), (45, 4), soprano_two_musicmaker_two],
492     [(45, 4), (91, 8), soprano_two_musicmaker_two],
493 ]
494 ])
495
496 voice_4_timespan_list = abjad.TimespanList([
497     abjad.AnnotatedTimespan(
498         start_offset=start_offset,
499         stop_offset=stop_offset,
500         annotation=MusicSpecifier(
501             music_maker=music_maker,
502             voice_name='Voice 4',
503         ),
504     )
505     for start_offset, stop_offset, music_maker in [
506         [(1, 8), (1, 4), soprano_three_musicmaker_two],
507         [(1, 4), (2, 4), soprano_three_musicmaker_two],
508         [(2, 4), (3, 4), soprano_three_musicmaker_two],
509         [(7, 4), (8, 4), soprano_three_musicmaker_two],
510         [(8, 4), (9, 4), soprano_three_musicmaker_two],
511         [(9, 4), (10, 4), soprano_three_musicmaker_two],
512         [(25, 8), (13, 4), soprano_three_musicmaker_one],
513         [(13, 4), (14, 4), soprano_three_musicmaker_one],
514         [(14, 4), (15, 4), soprano_three_musicmaker_one],
515         [(19, 4), (20, 4), soprano_three_musicmaker_two],
516         [(20, 4), (21, 4), soprano_three_musicmaker_two],
517         [(21, 4), (22, 4), soprano_three_musicmaker_two],
518         [(26, 4), (27, 4), soprano_three_musicmaker_one],
519         [(27, 4), (28, 4), soprano_three_musicmaker_one],
520         [(28, 4), (29, 4), soprano_three_musicmaker_one],
521         [(33, 4), (34, 4), soprano_three_musicmaker_one],
522         [(34, 4), (35, 4), soprano_three_musicmaker_one],
523         [(35, 4), (36, 4), soprano_three_musicmaker_one],
524         [(38, 4), (39, 4), soprano_three_musicmaker_two],
525         [(39, 4), (40, 4), soprano_three_musicmaker_two],
526         [(40, 4), (41, 4), soprano_three_musicmaker_two],
527         [(45, 4), (46, 4), soprano_three_musicmaker_one],
528         [(46, 4), (47, 4), soprano_three_musicmaker_one],
529         [(47, 4), (95, 8), soprano_three_musicmaker_one],
530     ]
531 ])
532
533 voice_5_timespan_list = abjad.TimespanList([
534     abjad.AnnotatedTimespan(
535         start_offset=start_offset,
536         stop_offset=stop_offset,
537         annotation=MusicSpecifier(
538             music_maker=music_maker,
539             voice_name='Voice 5',
540         ),
541     )
542     for start_offset, stop_offset, music_maker in [
543         [(0, 4), (1, 4), alto_one_musicmaker_one],
544         [(1, 4), (2, 4), alto_one_musicmaker_one],
545         [(2, 4), (3, 4), alto_one_musicmaker_one],
546     ]
547 ])

```

```

546     [(5, 4), (6, 4), alto_one_musicmaker_one],
547     [(6, 4), (7, 4), alto_one_musicmaker_one],
548     [(7, 4), (8, 4), alto_one_musicmaker_one],
549     [(12, 4), (13, 4), alto_one_musicmaker_two],
550     [(13, 4), (14, 4), alto_one_musicmaker_two],
551     [(14, 4), (15, 4), alto_one_musicmaker_two],
552     [(17, 4), (18, 4), alto_one_musicmaker_one],
553     [(18, 4), (19, 4), alto_one_musicmaker_one],
554     [(19, 4), (20, 4), alto_one_musicmaker_one],
555     [(28, 4), (29, 4), alto_one_musicmaker_two],
556     [(29, 4), (30, 4), alto_one_musicmaker_two],
557     [(30, 4), (31, 4), alto_one_musicmaker_two],
558     [(33, 4), (34, 4), alto_one_musicmaker_two],
559     [(34, 4), (35, 4), alto_one_musicmaker_two],
560     [(35, 4), (36, 4), alto_one_musicmaker_two],
561     [(40, 4), (41, 4), alto_one_musicmaker_one],
562     [(41, 4), (42, 4), alto_one_musicmaker_one],
563     [(42, 4), (43, 4), alto_one_musicmaker_one],
564     [(45, 4), (46, 4), alto_one_musicmaker_two],
565     [(46, 4), (47, 4), alto_one_musicmaker_two],
566     [(47, 4), (95, 8), alto_one_musicmaker_two],
567   ],
568 ])
569
570 voice_6_timespan_list = abjad.TimespanList([
571     abjad.AnnotatedTimespan(
572         start_offset=start_offset,
573         stop_offset=stop_offset,
574         annotation=MusicSpecifier(
575             music_maker=music_maker,
576             voice_name='Voice 6',
577         ),
578     ),
579     for start_offset, stop_offset, music_maker in [
580         [(4, 4), (5, 4), alto_two_musicmaker_two],
581         [(5, 4), (6, 4), alto_two_musicmaker_two],
582         [(6, 4), (7, 4), alto_two_musicmaker_two],
583         [(9, 4), (10, 4), alto_two_musicmaker_one],
584         [(10, 4), (11, 4), alto_two_musicmaker_one],
585         [(11, 4), (12, 4), alto_two_musicmaker_one],
586         [(16, 4), (17, 4), alto_two_musicmaker_two],
587         [(17, 4), (18, 4), alto_two_musicmaker_two],
588         [(18, 4), (19, 4), alto_two_musicmaker_two],
589         [(21, 4), (22, 4), alto_two_musicmaker_one],
590         [(22, 4), (23, 4), alto_two_musicmaker_one],
591         [(23, 4), (24, 4), alto_two_musicmaker_one],
592         [(24, 4), (25, 4), alto_two_musicmaker_one],
593         [(25, 4), (26, 4), alto_two_musicmaker_one],
594         [(26, 4), (27, 4), alto_two_musicmaker_one],
595         [(29, 4), (30, 4), alto_two_musicmaker_two],
596         [(30, 4), (31, 4), alto_two_musicmaker_two],
597         [(31, 4), (32, 4), alto_two_musicmaker_two],
598         [(36, 4), (37, 4), alto_two_musicmaker_one],
599         [(37, 4), (38, 4), alto_two_musicmaker_one],
600         [(38, 4), (39, 4), alto_two_musicmaker_one],
601         [(41, 4), (42, 4), alto_two_musicmaker_two],
602         [(42, 4), (43, 4), alto_two_musicmaker_two],
603         [(43, 4), (87, 8), alto_two_musicmaker_two],
604     ]
605   )
606 ]

```

```

605 ])
606
607 voice_7_timespan_list = abjad.TimespanList([
608     abjad.AnnotatedTimespan(
609         start_offset=start_offset,
610         stop_offset=stop_offset,
611         annotation=MusicSpecifier(
612             music_maker=music_maker,
613             voice_name='Voice 7',
614         ),
615     )
616     for start_offset, stop_offset, music_maker in [
617         [(2, 4), (3, 4), alto_three_musicmaker_one],
618         [(3, 4), (4, 4), alto_three_musicmaker_one],
619         [(4, 4), (5, 4), alto_three_musicmaker_one],
620         [(9, 4), (10, 4), alto_three_musicmaker_two],
621         [(10, 4), (11, 4), alto_three_musicmaker_two],
622         [(11, 4), (12, 4), alto_three_musicmaker_two],
623         [(14, 4), (15, 4), alto_three_musicmaker_two],
624         [(15, 4), (16, 4), alto_three_musicmaker_two],
625         [(16, 4), (17, 4), alto_three_musicmaker_two],
626         [(21, 4), (22, 4), alto_three_musicmaker_one],
627         [(22, 4), (23, 4), alto_three_musicmaker_one],
628         [(23, 4), (24, 4), alto_three_musicmaker_one],
629         [(24, 4), (25, 4), alto_three_musicmaker_two],
630         [(25, 4), (26, 4), alto_three_musicmaker_two],
631         [(26, 4), (27, 4), alto_three_musicmaker_two],
632         [(31, 4), (32, 4), alto_three_musicmaker_one],
633         [(32, 4), (33, 4), alto_three_musicmaker_one],
634         [(33, 4), (34, 4), alto_three_musicmaker_one],
635         [(36, 4), (37, 4), alto_three_musicmaker_one],
636         [(37, 4), (38, 4), alto_three_musicmaker_one],
637         [(38, 4), (39, 4), alto_three_musicmaker_one],
638         [(43, 4), (44, 4), alto_three_musicmaker_two],
639         [(44, 4), (45, 4), alto_three_musicmaker_two],
640         [(45, 4), (91, 8), alto_three_musicmaker_two],
641     ]
642 ])
643
644 voice_8_timespan_list = abjad.TimespanList([
645     abjad.AnnotatedTimespan(
646         start_offset=start_offset,
647         stop_offset=stop_offset,
648         annotation=MusicSpecifier(
649             music_maker=music_maker,
650             voice_name='Voice 8',
651         ),
652     )
653     for start_offset, stop_offset, music_maker in [
654         [(1, 8), (3, 4), alto_four_musicmaker_two],
655         [(7, 4), (9, 4), alto_four_musicmaker_two],
656         [(9, 4), (10, 4), alto_four_musicmaker_two],
657         [(25, 8), (13, 4), alto_four_musicmaker_one],
658         [(13, 4), (14, 4), alto_four_musicmaker_one],
659         [(14, 4), (15, 4), alto_four_musicmaker_one],
660         [(19, 4), (20, 4), alto_four_musicmaker_two],
661         [(20, 4), (21, 4), alto_four_musicmaker_two],
662         [(21, 4), (22, 4), alto_four_musicmaker_two],
663         [(26, 4), (27, 4), alto_four_musicmaker_one],

```

```

664     [(27, 4), (28, 4), alto_four_musicmaker_one],
665     [(28, 4), (29, 4), alto_four_musicmaker_one],
666     [(33, 4), (34, 4), alto_four_musicmaker_one],
667     [(34, 4), (35, 4), alto_four_musicmaker_one],
668     [(35, 4), (36, 4), alto_four_musicmaker_one],
669     [(38, 4), (39, 4), alto_four_musicmaker_two],
670     [(39, 4), (40, 4), alto_four_musicmaker_two],
671     [(40, 4), (41, 4), alto_four_musicmaker_two],
672     [(45, 4), (46, 4), alto_four_musicmaker_one],
673     [(46, 4), (47, 4), alto_four_musicmaker_one],
674     [(47, 4), (95, 8), alto_four_musicmaker_one],
675 ]
676 ])
677
678 voice_9_timespan_list = abjad.TimespanList([
679     abjad.AnnotatedTimespan(
680         start_offset=start_offset,
681         stop_offset=stop_offset,
682         annotation=MusicSpecifier(
683             music_maker=music_maker,
684             voice_name='Voice 9',
685         ),
686     )
687     for start_offset, stop_offset, music_maker in [
688         [(0, 4), (1, 4), alto_five_musicmaker_one],
689         [(1, 4), (3, 4), alto_five_musicmaker_one],
690         [(5, 4), (7, 4), alto_five_musicmaker_one],
691         [(7, 4), (8, 4), alto_five_musicmaker_one],
692         [(12, 4), (13, 4), alto_five_musicmaker_two],
693         [(13, 4), (15, 4), alto_five_musicmaker_two],
694         [(17, 4), (19, 4), alto_five_musicmaker_one],
695         [(19, 4), (20, 4), alto_five_musicmaker_one],
696         [(28, 4), (29, 4), alto_five_musicmaker_two],
697         [(29, 4), (31, 4), alto_five_musicmaker_two],
698         [(33, 4), (35, 4), alto_five_musicmaker_two],
699         [(35, 4), (36, 4), alto_five_musicmaker_two],
700         [(40, 4), (41, 4), alto_five_musicmaker_one],
701         [(41, 4), (42, 4), alto_five_musicmaker_one],
702         [(42, 4), (43, 4), alto_five_musicmaker_one],
703         [(45, 4), (46, 4), alto_five_musicmaker_two],
704         [(46, 4), (47, 4), alto_five_musicmaker_two],
705         [(47, 4), (95, 8), alto_five_musicmaker_two],
706     ]
707 ])
708
709 voice_10_timespan_list = abjad.TimespanList([
710     abjad.AnnotatedTimespan(
711         start_offset=start_offset,
712         stop_offset=stop_offset,
713         annotation=MusicSpecifier(
714             music_maker=music_maker,
715             voice_name='Voice 10',
716         ),
717     )
718     for start_offset, stop_offset, music_maker in [
719         [(4, 4), (5, 4), alto_six_musicmaker_two],
720         [(5, 4), (7, 4), alto_six_musicmaker_two],
721         [(9, 4), (11, 4), alto_six_musicmaker_one],
722         [(11, 4), (12, 4), alto_six_musicmaker_one],

```

```

723     [(16, 4), (17, 4), alto_six_musicmaker_two],
724     [(17, 4), (19, 4), alto_six_musicmaker_two],
725     [(21, 4), (23, 4), alto_six_musicmaker_one],
726     [(23, 4), (24, 4), alto_six_musicmaker_one],
727     [(24, 4), (26, 4), alto_six_musicmaker_one],
728     [(26, 4), (27, 4), alto_six_musicmaker_one],
729     [(29, 4), (31, 4), alto_six_musicmaker_two],
730     [(31, 4), (32, 4), alto_six_musicmaker_two],
731     [(36, 4), (38, 4), alto_six_musicmaker_one],
732     [(38, 4), (39, 4), alto_six_musicmaker_one],
733     [(41, 4), (42, 4), alto_six_musicmaker_two],
734     [(42, 4), (43, 4), alto_six_musicmaker_two],
735     [(43, 4), (87, 8), alto_six_musicmaker_two],
736 ]
737 ])
738
739 voice_11_timespan_list = abjad.TimespanList([
740     abjad.AnnotatedTimespan(
741         start_offset=start_offset,
742         stop_offset=stop_offset,
743         annotation=MusicSpecifier(
744             music_maker=music_maker,
745             voice_name='Voice 11',
746         ),
747     )
748     for start_offset, stop_offset, music_maker in [
749         [(2, 4), (4, 4), tenor_one_musicmaker_one],
750         [(4, 4), (5, 4), tenor_one_musicmaker_one],
751         [(9, 4), (10, 4), tenor_one_musicmaker_two],
752         [(10, 4), (12, 4), tenor_one_musicmaker_two],
753         [(14, 4), (16, 4), tenor_one_musicmaker_two],
754         [(16, 4), (17, 4), tenor_one_musicmaker_two],
755         [(21, 4), (22, 4), tenor_one_musicmaker_one],
756         [(22, 4), (24, 4), tenor_one_musicmaker_one],
757         [(24, 4), (25, 4), tenor_one_musicmaker_two],
758         [(25, 4), (26, 4), tenor_one_musicmaker_two],
759         [(26, 4), (27, 4), tenor_one_musicmaker_two],
760         [(31, 4), (33, 4), tenor_one_musicmaker_one],
761         [(33, 4), (34, 4), tenor_one_musicmaker_one],
762         [(36, 4), (37, 4), tenor_one_musicmaker_one],
763         [(37, 4), (39, 4), tenor_one_musicmaker_one],
764         [(43, 4), (45, 4), tenor_one_musicmaker_two],
765         [(45, 4), (91, 8), tenor_one_musicmaker_two],
766     ]
767 ])
768
769 voice_12_timespan_list = abjad.TimespanList([
770     abjad.AnnotatedTimespan(
771         start_offset=start_offset,
772         stop_offset=stop_offset,
773         annotation=MusicSpecifier(
774             music_maker=music_maker,
775             voice_name='Voice 12',
776         ),
777     )
778     for start_offset, stop_offset, music_maker in [
779         [(1, 8), (2, 4), tenor_two_musicmaker_two],
780         [(2, 4), (3, 4), tenor_two_musicmaker_two],
781         [(7, 4), (8, 4), tenor_two_musicmaker_two],

```

```

782     [(8, 4), (9, 4), tenor_two_musicmaker_two],
783     [(9, 4), (10, 4), tenor_two_musicmaker_two],
784     [(25, 8), (14, 4), tenor_two_musicmaker_one],
785     [(14, 4), (15, 4), tenor_two_musicmaker_one],
786     [(19, 4), (20, 4), tenor_two_musicmaker_two],
787     [(20, 4), (21, 4), tenor_two_musicmaker_two],
788     [(21, 4), (22, 4), tenor_two_musicmaker_two],
789     [(26, 4), (27, 4), tenor_two_musicmaker_one],
790     [(27, 4), (28, 4), tenor_two_musicmaker_one],
791     [(28, 4), (29, 4), tenor_two_musicmaker_one],
792     [(33, 4), (34, 4), tenor_two_musicmaker_one],
793     [(34, 4), (36, 4), tenor_two_musicmaker_one],
794     [(38, 4), (39, 4), tenor_two_musicmaker_two],
795     [(39, 4), (40, 4), tenor_two_musicmaker_two],
796     [(40, 4), (41, 4), tenor_two_musicmaker_two],
797     [(45, 4), (47, 4), tenor_two_musicmaker_one],
798     [(47, 4), (95, 8), tenor_two_musicmaker_one],
799   ],
800 ])
801
802 voice_13_timespan_list = abjad.TimespanList([
803     abjad.AnnotatedTimespan(
804         start_offset=start_offset,
805         stop_offset=stop_offset,
806         annotation=MusicSpecifier(
807             music_maker=music_maker,
808             voice_name='Voice 13',
809         ),
810     ),
811     for start_offset, stop_offset, music_maker in [
812         [(0, 4), (1, 4), tenor_three_musicmaker_one],
813         [(1, 4), (3, 4), tenor_three_musicmaker_one],
814         [(5, 4), (7, 4), tenor_three_musicmaker_one],
815         [(7, 4), (8, 4), tenor_three_musicmaker_one],
816         [(12, 4), (13, 4), tenor_three_musicmaker_two],
817         [(13, 4), (15, 4), tenor_three_musicmaker_two],
818         [(17, 4), (19, 4), tenor_three_musicmaker_one],
819         [(19, 4), (20, 4), tenor_three_musicmaker_one],
820         [(28, 4), (29, 4), tenor_three_musicmaker_two],
821         [(29, 4), (31, 4), tenor_three_musicmaker_two],
822         [(33, 4), (35, 4), tenor_three_musicmaker_two],
823         [(35, 4), (36, 4), tenor_three_musicmaker_two],
824         [(40, 4), (41, 4), tenor_three_musicmaker_one],
825         [(41, 4), (43, 4), tenor_three_musicmaker_one],
826         [(45, 4), (46, 4), tenor_three_musicmaker_two],
827         [(46, 4), (47, 4), tenor_three_musicmaker_two],
828         [(47, 4), (95, 8), tenor_three_musicmaker_two],
829     ],
830   ])
831
832 voice_14_timespan_list = abjad.TimespanList([
833     abjad.AnnotatedTimespan(
834         start_offset=start_offset,
835         stop_offset=stop_offset,
836         annotation=MusicSpecifier(
837             music_maker=music_maker,
838             voice_name='Voice 14',
839         ),
840   )

```

```

841     for start_offset, stop_offset, music_maker in [
842         [(4, 4), (5, 4), tenor_four_musicmaker_two],
843         [(5, 4), (6, 4), tenor_four_musicmaker_two],
844         [(6, 4), (7, 4), tenor_four_musicmaker_two],
845         [(9, 4), (10, 4), tenor_four_musicmaker_one],
846         [(10, 4), (11, 4), tenor_four_musicmaker_one],
847         [(11, 4), (12, 4), tenor_four_musicmaker_one],
848         [(16, 4), (17, 4), tenor_four_musicmaker_two],
849         [(17, 4), (18, 4), tenor_four_musicmaker_two],
850         [(18, 4), (19, 4), tenor_four_musicmaker_two],
851         [(21, 4), (22, 4), tenor_four_musicmaker_one],
852         [(22, 4), (24, 4), tenor_four_musicmaker_one],
853         [(24, 4), (26, 4), tenor_four_musicmaker_one],
854         [(26, 4), (27, 4), tenor_four_musicmaker_one],
855         [(29, 4), (30, 4), tenor_four_musicmaker_two],
856         [(30, 4), (31, 4), tenor_four_musicmaker_two],
857         [(31, 4), (32, 4), tenor_four_musicmaker_two],
858         [(36, 4), (38, 4), tenor_four_musicmaker_one],
859         [(38, 4), (39, 4), tenor_four_musicmaker_one],
860         [(41, 4), (42, 4), tenor_four_musicmaker_two],
861         [(42, 4), (43, 4), tenor_four_musicmaker_two],
862         [(43, 4), (87, 8), tenor_four_musicmaker_two],
863     ]
864 ])
865
866 voice_15_timespan_list = abjad.TimespanList([
867     abjad.AnnotatedTimespan(
868         start_offset=start_offset,
869         stop_offset=stop_offset,
870         annotation=MusicSpecifier(
871             music_maker=music_maker,
872             voice_name='Voice 15',
873         ),
874     )
875     for start_offset, stop_offset, music_maker in [
876         [(2, 4), (3, 4), tenor_five_musicmaker_one],
877         [(3, 4), (5, 4), tenor_five_musicmaker_one],
878         [(9, 4), (11, 4), tenor_five_musicmaker_two],
879         [(11, 4), (12, 4), tenor_five_musicmaker_two],
880         [(14, 4), (15, 4), tenor_five_musicmaker_two],
881         [(15, 4), (17, 4), tenor_five_musicmaker_two],
882         [(21, 4), (23, 4), tenor_five_musicmaker_one],
883         [(23, 4), (24, 4), tenor_five_musicmaker_one],
884         [(24, 4), (26, 4), tenor_five_musicmaker_two],
885         [(26, 4), (27, 4), tenor_five_musicmaker_two],
886         [(31, 4), (32, 4), tenor_five_musicmaker_one],
887         [(32, 4), (34, 4), tenor_five_musicmaker_one],
888         [(36, 4), (38, 4), tenor_five_musicmaker_one],
889         [(38, 4), (39, 4), tenor_five_musicmaker_one],
890         [(43, 4), (44, 4), tenor_five_musicmaker_two],
891         [(44, 4), (45, 4), tenor_five_musicmaker_two],
892         [(45, 4), (91, 8), tenor_five_musicmaker_two],
893     ]
894 ])
895
896 voice_16_timespan_list = abjad.TimespanList([
897     abjad.AnnotatedTimespan(
898         start_offset=start_offset,
899         stop_offset=stop_offset,

```

```

900     annotation=MusicSpecifier(
901         music_maker=music_maker,
902         voice_name='Voice 16',
903     ),
904 )
905 for start_offset, stop_offset, music_maker in [
906     [(1, 8), (2, 4), baritone_one_musicmaker_two],
907     [(2, 4), (3, 4), baritone_one_musicmaker_two],
908     [(7, 4), (8, 4), baritone_one_musicmaker_two],
909     [(8, 4), (9, 4), baritone_one_musicmaker_two],
910     [(9, 4), (10, 4), baritone_one_musicmaker_two],
911     [(25, 8), (13, 4), baritone_one_musicmaker_one],
912     [(13, 4), (15, 4), baritone_one_musicmaker_one],
913     [(19, 4), (21, 4), baritone_one_musicmaker_two],
914     [(21, 4), (22, 4), baritone_one_musicmaker_two],
915     [(26, 4), (27, 4), baritone_one_musicmaker_one],
916     [(27, 4), (29, 4), baritone_one_musicmaker_one],
917     [(33, 4), (35, 4), baritone_one_musicmaker_one],
918     [(35, 4), (36, 4), baritone_one_musicmaker_one],
919     [(38, 4), (39, 4), baritone_one_musicmaker_two],
920     [(39, 4), (41, 4), baritone_one_musicmaker_two],
921     [(45, 4), (46, 4), baritone_one_musicmaker_one],
922     [(46, 4), (47, 4), baritone_one_musicmaker_one],
923     [(47, 4), (95, 8), baritone_one_musicmaker_one],
924 ]
925 ])
926
927 voice_17_timespan_list = abjad.TimespanList([
928     abjad.AnnotatedTimespan(
929         start_offset=start_offset,
930         stop_offset=stop_offset,
931         annotation=MusicSpecifier(
932             music_maker=music_maker,
933             voice_name='Voice 17',
934         ),
935     ),
936     for start_offset, stop_offset, music_maker in [
937         [(0, 4), (1, 4), baritone_two_musicmaker_one],
938         [(1, 4), (2, 4), baritone_two_musicmaker_one],
939         [(2, 4), (3, 4), baritone_two_musicmaker_one],
940         [(5, 4), (6, 4), baritone_two_musicmaker_one],
941         [(6, 4), (8, 4), baritone_two_musicmaker_one],
942         [(12, 4), (14, 4), baritone_two_musicmaker_two],
943         [(14, 4), (15, 4), baritone_two_musicmaker_two],
944         [(17, 4), (18, 4), baritone_two_musicmaker_one],
945         [(18, 4), (20, 4), baritone_two_musicmaker_one],
946         [(28, 4), (30, 4), baritone_two_musicmaker_two],
947         [(30, 4), (31, 4), baritone_two_musicmaker_two],
948         [(33, 4), (34, 4), baritone_two_musicmaker_two],
949         [(34, 4), (36, 4), baritone_two_musicmaker_two],
950         [(40, 4), (42, 4), baritone_two_musicmaker_one],
951         [(42, 4), (43, 4), baritone_two_musicmaker_one],
952         [(45, 4), (47, 4), baritone_two_musicmaker_two],
953         [(47, 4), (95, 8), baritone_two_musicmaker_two],
954     ]
955 ])
956
957 voice_18_timespan_list = abjad.TimespanList([
958     abjad.AnnotatedTimespan(

```

```

959         start_offset=start_offset,
960         stop_offset=stop_offset,
961         annotation=MusicSpecifier(
962             music_maker=music_maker,
963             voice_name='Voice 18',
964         ),
965     ),
966     for start_offset, stop_offset, music_maker in [
967         [(4, 4), (5, 4), baritone_three_musicmaker_two],
968         [(5, 4), (7, 4), baritone_three_musicmaker_two],
969         [(9, 4), (11, 4), baritone_three_musicmaker_one],
970         [(11, 4), (12, 4), baritone_three_musicmaker_one],
971         [(16, 4), (17, 4), baritone_three_musicmaker_two],
972         [(17, 4), (19, 4), baritone_three_musicmaker_two],
973         [(21, 4), (23, 4), baritone_three_musicmaker_one],
974         [(23, 4), (24, 4), baritone_three_musicmaker_one],
975         [(24, 4), (25, 4), baritone_three_musicmaker_one],
976         [(25, 4), (27, 4), baritone_three_musicmaker_one],
977         [(29, 4), (31, 4), baritone_three_musicmaker_two],
978         [(31, 4), (32, 4), baritone_three_musicmaker_two],
979         [(36, 4), (37, 4), baritone_three_musicmaker_one],
980         [(37, 4), (39, 4), baritone_three_musicmaker_one],
981         [(41, 4), (43, 4), baritone_three_musicmaker_two],
982         [(43, 4), (87, 8), baritone_three_musicmaker_two],
983     ]
984 ])
985
986 voice_19_timespan_list = abjad.TimespanList([
987     abjad.AnnotatedTimespan(
988         start_offset=start_offset,
989         stop_offset=stop_offset,
990         annotation=MusicSpecifier(
991             music_maker=music_maker,
992             voice_name='Voice 19',
993         ),
994     ),
995     for start_offset, stop_offset, music_maker in [
996         [(2, 4), (4, 4), bass_one_musicmaker_one],
997         [(4, 4), (5, 4), bass_one_musicmaker_one],
998         [(9, 4), (11, 4), bass_one_musicmaker_two],
999         [(11, 4), (12, 4), bass_one_musicmaker_two],
1000         [(14, 4), (15, 4), bass_one_musicmaker_two],
1001         [(15, 4), (17, 4), bass_one_musicmaker_two],
1002         [(21, 4), (22, 4), bass_one_musicmaker_one],
1003         [(22, 4), (24, 4), bass_one_musicmaker_one],
1004         [(24, 4), (25, 4), bass_one_musicmaker_two],
1005         [(25, 4), (26, 4), bass_one_musicmaker_two],
1006         [(26, 4), (27, 4), bass_one_musicmaker_two],
1007         [(31, 4), (33, 4), bass_one_musicmaker_one],
1008         [(33, 4), (34, 4), bass_one_musicmaker_one],
1009         [(36, 4), (37, 4), bass_one_musicmaker_one],
1010         [(37, 4), (39, 4), bass_one_musicmaker_one],
1011         [(43, 4), (44, 4), bass_one_musicmaker_two],
1012         [(44, 4), (91, 8), bass_one_musicmaker_two],
1013     ]
1014 ])
1015
1016 voice_20_timespan_list = abjad.TimespanList([
1017     abjad.AnnotatedTimespan(

```

```

1018     start_offset=start_offset,
1019     stop_offset=stop_offset,
1020     annotation=MusicSpecifier(
1021         music_maker=music_maker,
1022         voice_name='Voice 20',
1023     ),
1024 )
1025 for start_offset, stop_offset, music_maker in [
1026     [(1, 8), (1, 4), bass_two_musicmaker_two],
1027     [(1, 4), (2, 4), bass_two_musicmaker_two],
1028     [(2, 4), (3, 4), bass_two_musicmaker_two],
1029     [(7, 4), (8, 4), bass_two_musicmaker_two],
1030     [(8, 4), (9, 4), bass_two_musicmaker_two],
1031     [(9, 4), (10, 4), bass_two_musicmaker_two],
1032     [(25, 8), (13, 4), bass_two_musicmaker_one],
1033     [(13, 4), (14, 4), bass_two_musicmaker_one],
1034     [(14, 4), (15, 4), bass_two_musicmaker_one],
1035     [(19, 4), (20, 4), bass_two_musicmaker_two],
1036     [(20, 4), (21, 4), bass_two_musicmaker_two],
1037     [(21, 4), (22, 4), bass_two_musicmaker_two],
1038     [(26, 4), (27, 4), bass_two_musicmaker_one],
1039     [(27, 4), (28, 4), bass_two_musicmaker_one],
1040     [(28, 4), (29, 4), bass_two_musicmaker_one],
1041     [(33, 4), (34, 4), bass_two_musicmaker_one],
1042     [(34, 4), (35, 4), bass_two_musicmaker_one],
1043     [(35, 4), (36, 4), bass_two_musicmaker_one],
1044     [(38, 4), (39, 4), bass_two_musicmaker_two],
1045     [(39, 4), (40, 4), bass_two_musicmaker_two],
1046     [(40, 4), (41, 4), bass_two_musicmaker_two],
1047     [(45, 4), (46, 4), bass_two_musicmaker_one],
1048     [(46, 4), (47, 4), bass_two_musicmaker_one],
1049     [(47, 4), (95, 8), bass_two_musicmaker_one],
1050 ]
1051 ])
1052
1053 voice_21_timespan_list = abjad.TimespanList([
1054     abjad.AnnotatedTimespan(
1055         start_offset=start_offset,
1056         stop_offset=stop_offset,
1057         annotation=MusicSpecifier(
1058             music_maker=music_maker,
1059             voice_name='Voice 21',
1060         ),
1061     ),
1062     for start_offset, stop_offset, music_maker in [
1063         [(0, 4), (2, 4), contrabass_musicmaker_one],
1064         [(2, 4), (3, 4), contrabass_musicmaker_one],
1065         [(5, 4), (7, 4), contrabass_musicmaker_one],
1066         [(7, 4), (8, 4), contrabass_musicmaker_one],
1067         [(12, 4), (14, 4), contrabass_musicmaker_two],
1068         [(14, 4), (15, 4), contrabass_musicmaker_two],
1069         [(17, 4), (18, 4), contrabass_musicmaker_one],
1070         [(18, 4), (20, 4), contrabass_musicmaker_one],
1071         [(28, 4), (31, 4), contrabass_musicmaker_two],
1072         [(33, 4), (35, 4), contrabass_musicmaker_two],
1073         [(35, 4), (36, 4), contrabass_musicmaker_two],
1074         [(40, 4), (42, 4), contrabass_musicmaker_one],
1075         [(42, 4), (43, 4), contrabass_musicmaker_one],
1076         [(45, 4), (46, 4), contrabass_musicmaker_two],

```

```

1077     [(46, 4), (47, 4), contrabass_musicmaker_two],
1078     [(47, 4), (95, 8), contrabass_musicmaker_two],
1079 ]
1080 ])
1081
1082 all_timespan_lists = {
1083     'Voice 1': voice_1_timespan_list,
1084     'Voice 2': voice_2_timespan_list,
1085     'Voice 3': voice_3_timespan_list,
1086     'Voice 4': voice_4_timespan_list,
1087     'Voice 5': voice_5_timespan_list,
1088     'Voice 6': voice_6_timespan_list,
1089     'Voice 7': voice_7_timespan_list,
1090     'Voice 8': voice_8_timespan_list,
1091     'Voice 9': voice_9_timespan_list,
1092     'Voice 10': voice_10_timespan_list,
1093     'Voice 11': voice_11_timespan_list,
1094     'Voice 12': voice_12_timespan_list,
1095     'Voice 13': voice_13_timespan_list,
1096     'Voice 14': voice_14_timespan_list,
1097     'Voice 15': voice_15_timespan_list,
1098     'Voice 16': voice_16_timespan_list,
1099     'Voice 17': voice_17_timespan_list,
1100     'Voice 18': voice_18_timespan_list,
1101     'Voice 19': voice_19_timespan_list,
1102     'Voice 20': voice_20_timespan_list,
1103     'Voice 21': voice_21_timespan_list,
1104 }
1105
1106 global_timespan = abjad.Timespan(
1107     start_offset=0,
1108     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
1109 )
1110
1111 for voice_name, timespan_list in all_timespan_lists.items():
1112     silences = abjad.TimespanList([global_timespan])
1113     silences.extend(timespan_list)
1114     silences.sort()
1115     silences.compute_logical_xor()
1116     for silence_timespan in silences:
1117         timespan_list.append(
1118             abjad.AnnotatedTimespan(
1119                 start_offset=silence_timespan.start_offset,
1120                 stop_offset=silence_timespan.stop_offset,
1121                 annotation=MusicSpecifier(
1122                     music_maker=None,
1123                     voice_name=voice_name,
1124                     ),
1125                 )
1126             )
1127     timespan_list.sort()
1128
1129 for voice_name, timespan_list in all_timespan_lists.items():
1130     shards = timespan_list.split_at_offsets(bounds)
1131     split_timespan_list = abjad.TimespanList()
1132     for shard in shards:
1133         split_timespan_list.extend(shard)
1134     split_timespan_list.sort()
1135     all_timespan_lists[voice_name] = timespan_list

```

```

1136 score = abjad.Score([
1137     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
1138     abjad.StaffGroup(
1139         [
1140             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
1141             lilypond_type='Staff'),
1142             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
1143             lilypond_type='Staff'),
1144             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
1145             lilypond_type='Staff'),
1146             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
1147             lilypond_type='Staff'),
1148             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
1149             lilypond_type='Staff'),
1150             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
1151             lilypond_type='Staff'),
1152             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
1153             lilypond_type='Staff'),
1154             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
1155             lilypond_type='Staff'),
1156             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
1157             lilypond_type='Staff'),
1158             abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
1159             lilypond_type='Staff'),
1160             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
1161             lilypond_type='Staff'),
1162             abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
1163             lilypond_type='Staff'),
1164             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
1165             lilypond_type='Staff'),
1166             abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
1167             lilypond_type='Staff'),
1168             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
1169             lilypond_type='Staff'),
1170             abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
1171             lilypond_type='Staff'),
1172             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
1173             lilypond_type='Staff'),
1174             abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
1175             lilypond_type='Staff'),
1176             abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
1177             lilypond_type='Staff'),
1178             abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
1179             lilypond_type='Staff'),
1180             abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
1181             lilypond_type='Staff'),
1182         ],
1183         name='Staff Group',
1184     )
1185 ])
1186
1187 for time_signature in time_signatures:
1188     skip = abjad.Skip(1, multiplier=(time_signature))
1189     abjad.attach(time_signature, skip)
1190     score['Global Context'].append(skip)
1191
1192 print('Making containers ...')
1193

```

```

1174 def make_container(music_maker, durations):
1175     selections = music_maker(durations)
1176     container = abjad.Container([])
1177     container.extend(selections)
1178     return container
1179
1180 def key_function(timespan):
1181     return timespan.annotation.music_maker or silence_maker
1182
1183 for voice_name, timespan_list in all_timespan_lists.items():
1184     for music_maker, grouper in itertools.groupby(
1185         timespan_list,
1186         key=key_function,
1187     ):
1188         durations = [timespan.duration for timespan in grouper]
1189         container = make_container(music_maker, durations)
1190         voice = score[voice_name]
1191         voice.append(container)
1192
1193 print('Splitting and rewriting ...')
1194 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
1195     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
1196         time_signature = time_signatures[i]
1197         abjad.mutate(shard).rewrite_meter(time_signature)
1198
1199 print('Beaming runs ...')
1200 for voice in abjad.select(score).components(abjad.Voice):
1201     for run in abjad.select(voice).runs():
1202         specifier = abjadext.rmakers.BeamSpecifier(
1203             beam_each_division=False,
1204         )
1205         specifier(run)
1206         abjad.beam(voice[:, beam_lone_notes=False, beam_rests=False])
1207
1208 print('Stopping Hairpins ...')
1209 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1210     for rest in abjad.iterate(staff).components(abjad.Rest):
1211         previous_leaf = abjad.inspect(rest).leaf(-1)
1212         if isinstance(previous_leaf, abjad.Note):
1213             abjad.attach(abjad.StopHairpin(), rest)
1214         elif isinstance(previous_leaf, abjad.Chord):
1215             abjad.attach(abjad.StopHairpin(), rest)
1216         elif isinstance(previous_leaf, abjad.Rest):
1217             pass
1218
1219 scales = [
1220     soprano_1_notes,
1221     soprano_2_notes,
1222     soprano_3_notes,
1223     alto_1_notes,
1224     alto_2_notes,
1225     alto_3_notes,
1226     alto_4_notes,
1227     alto_5_notes,
1228     alto_6_notes,
1229     tenor_1_notes,
1230     tenor_2_notes,
1231     tenor_3_notes,
1232 ]

```

```

1233     tenor_4_notes,
1234     tenor_5_notes,
1235     baritone_1_notes,
1236     baritone_2_notes,
1237     baritone_3_notes,
1238     bass_1_notes,
1239     bass_2_notes,
1240     contrabass_notes,
1241 ]
1242
1243 staffs = [staff for staff in abjad.iterate(score['Staff Group']).components(abjad.
1244     Staff)]
1245
1246 for staff, scale in zip(staffs, scales):
1247     logicl_ties = [i for i in abjad.iterate(staff).logical_ties(pitched=True)]
1248     pitches = cyc(scale)
1249     for i, logicl_tie in enumerate(logicl_ties):
1250         if logicl_tie.is_pitched == True:
1251             pitch = next(pitches)
1252             for note in logicl_tie:
1253                 note.written_pitch = pitch
1254
1255 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1256     notes = abjad.select(staff).logical_ties(pitched=True)
1257     abjad.glissando(
1258         notes[0:24],
1259         allow_repeats=True,
1260     )
1261     abjad.glissando(
1262         notes[24:],
1263         allow_repeats=True,
1264     )
1265 #attach instruments and clefs
1266
1267 print('Adding attachments ...')
1268 bar_line = abjad.BarLine('||')
1269 metro = abjad.MetronomeMark((1, 4), 60)
1270 markup = abjad.Markup(r'\bold { Invocation }')
1271 mark = abjad.RehearsalMark(markup=markup)
1272
1273 instruments = cyc([
1274     abjad.SopranoSaxophone(),
1275     abjad.SopranoSaxophone(),
1276     abjad.SopranoSaxophone(),
1277     abjad.SopranoSaxophone(),
1278     abjad.AltoSaxophone(),
1279     abjad.AltoSaxophone(),
1280     abjad.AltoSaxophone(),
1281     abjad.AltoSaxophone(),
1282     abjad.AltoSaxophone(),
1283     abjad.AltoSaxophone(),
1284     abjad.TenorSaxophone(),
1285     abjad.TenorSaxophone(),
1286     abjad.TenorSaxophone(),
1287     abjad.TenorSaxophone(),
1288     abjad.TenorSaxophone(),
1289     abjad.BaritoneSaxophone(),
1290     abjad.BaritoneSaxophone(),

```

```

1291 abjad.BaritoneSaxophone(),
1292 abjad.BassSaxophone(),
1293 abjad.BassSaxophone(),
1294 abjad.ContrabassSaxophone(),
1295 ])
1296
1297 abbreviations = cyc([
1298     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
1299     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
1300     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
1301     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
1302     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
1303     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
1304     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
1305     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
1306     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
1307     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
1308     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
1309     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
1310     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
1311     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
1312     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
1313     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
1314     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
1315     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
1316     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
1317     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
1318     abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
1319 ])
1320
1321 names = cyc([
1322     abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
1323     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
1324     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
1325     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
1326     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
1327     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
1328     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
1329     abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
1330     abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
1331     abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
1332     abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
1333     abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
1334     abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
1335     abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
1336     abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
1337     abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
1338     abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
1339     abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
1340     abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
1341     abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
1342     abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
1343 ])
1344
1345 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1346     leaf1 = abjad.select(staff).leaves()[0]
1347     abjad.attach(next(instruments), leaf1)
1348     abjad.attach(next(abbreviations), leaf1)
1349     abjad.attach(next(names), leaf1)

```

```

1350
1351 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
1352     leaf1 = abjad.select(staff).leaves()[0]
1353     last_leaf = abjad.select(staff).leaves()[-1]
1354     abjad.attach(metro, leaf1)
1355     abjad.attach(bar_line, last_leaf)
1356
1357 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
1358     leaf1 = abjad.select(staff).leaves()[0]
1359     abjad.attach(mark, leaf1)
1360
1361 score_file = abjad.LilyPondFile.new(
1362     score,
1363     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1364     _stylesheets/abjad.ily'],
1365 )
1366
1367 abjad.SegmentMaker.comment_measure_numbers(score)
1368 ######
1369
1370 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Invocation'
1371 pdf_path = f'{directory}/Invocation.pdf'
1372 path = pathlib.Path('Invocation.pdf')
1373 if path.exists():
1374     print(f'Removing {pdf_path} ...')
1375     path.unlink()
1376 time_1 = time.time()
1377 print(f'Persisting {pdf_path} ...')
1378 result = abjad.persist(score_file).as_pdf(pdf_path)
1379 print(result[0])
1380 print(result[1])
1381 print(result[2])
1382 success = result[3]
1383 if success is False:
1384     print('LilyPond failed!')
1385 time_2 = time.time()
1386 total_time = time_2 - time_1
1387 print(f'Total time: {total_time} seconds')
1388 if path.exists():
1389     print(f'Opening {pdf_path} ...')
1390     os.system(f'open {pdf_path}')
1391 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Invocation/
1392     Invocation.ly').readlines()
1393 open('/Users/evansdsg2/Scores/guerrero/Build/Invocation.ly', 'w').writelines(
1394     score_lines[15:-1])
1395
1396 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
1397     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1398     signature_copy = abjad.mutate(signatures).copy()
1399     staff_copy = abjad.mutate(staff).copy()
1400     part = abjad.Score()
1401     part.insert(0, staff)
1402     part.insert(0, signature_copy)
1403     part_file = abjad.LilyPondFile.new(
1404         part,
1405         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1406         _stylesheets/abjad.ily'],
1407     )
1408     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano

```

```

1405 pdf_path = f'{directory}/Invocation.pdf'
1406 path = pathlib.Path('Invocation.pdf')
1407 if path.exists():
1408     print(f'Removing {pdf_path} ...')
1409     path.unlink()
1410 time_1 = time.time()
1411 print(f'Persisting {pdf_path} ...')
1412 result = abjad.persist(part_file).as_pdf(pdf_path)
1413 print(result[0])
1414 print(result[1])
1415 print(result[2])
1416 success = result[3]
1417 if success is False:
1418     print('LilyPond failed!')
1419 time_2 = time.time()
1420 total_time = time_2 - time_1
1421 print(f'Total time: {total_time} seconds')
1422 if path.exists():
1423     print(f'Opening {pdf_path} ...')
1424     os.system(f'open {pdf_path}')
1425 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano1/
Invocation.ly').readlines()
1426 open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano1/Invocation.ly',
'w').writelines(part_lines[15:-1])
1427
1428 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
1429     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1430     signature_copy = abjad.mutate(signatures).copy()
1431     staff_copy = abjad.mutate(staff).copy()
1432     part = abjad.Score()
1433     part.insert(0, staff)
1434     part.insert(0, signature_copy)
1435     part_file = abjad.LilyPondFile.new(
1436         part,
1437         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1438         )
1439 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1'
1440 pdf_path = f'{directory}/Invocation.pdf'
1441 path = pathlib.Path('Invocation.pdf')
1442 if path.exists():
1443     print(f'Removing {pdf_path} ...')
1444     path.unlink()
1445 time_1 = time.time()
1446 print(f'Persisting {pdf_path} ...')
1447 result = abjad.persist(part_file).as_pdf(pdf_path)
1448 print(result[0])
1449 print(result[1])
1450 print(result[2])
1451 success = result[3]
1452 if success is False:
1453     print('LilyPond failed!')
1454 time_2 = time.time()
1455 total_time = time_2 - time_1
1456 print(f'Total time: {total_time} seconds')
1457 if path.exists():
1458     print(f'Opening {pdf_path} ...')
1459     os.system(f'open {pdf_path}')
1460 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/

```

```

1461     Invocation.ly').readlines()
1462     open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/Invocation.ly',
1463         'w').writelines(part_lines[15:-1])
1464
1465 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
1466     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1467     signature_copy = abjad.mutate(signatures).copy()
1468     staff_copy = abjad.mutate(staff).copy()
1469     part = abjad.Score()
1470     part.insert(0, staff)
1471     part.insert(0, signature_copy)
1472     part_file = abjad.LilyPondFile.new(
1473         part,
1474         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1475         _stylesheets/abjad.ily'],
1476         )
1477     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2'
1478     pdf_path = f'{directory}/Invocation.pdf'
1479     path = pathlib.Path('Invocation.pdf')
1480     if path.exists():
1481         print(f'Removing {pdf_path} ...')
1482         path.unlink()
1483     time_1 = time.time()
1484     print(f'Persisting {pdf_path} ...')
1485     result = abjad.persist(part_file).as_pdf(pdf_path)
1486     print(result[0])
1487     print(result[1])
1488     print(result[2])
1489     success = result[3]
1490     if success is False:
1491         print('LilyPond failed!')
1492     time_2 = time.time()
1493     total_time = time_2 - time_1
1494     print(f'Total time: {total_time} seconds')
1495     if path.exists():
1496         print(f'Opening {pdf_path} ...')
1497         os.system(f'open {pdf_path}')
1498     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/
1499     Invocation.ly').readlines()
1500     open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/Invocation.ly',
1501         'w').writelines(part_lines[15:-1])
1502
1503 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
1504     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1505     signature_copy = abjad.mutate(signatures).copy()
1506     staff_copy = abjad.mutate(staff).copy()
1507     part = abjad.Score()
1508     part.insert(0, staff)
1509     part.insert(0, signature_copy)
1510     part_file = abjad.LilyPondFile.new(
1511         part,
1512         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1513         _stylesheets/abjad.ily'],
1514         )
1515     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3'
1516     pdf_path = f'{directory}/Invocation.pdf'
1517     path = pathlib.Path('Invocation.pdf')
1518     if path.exists():
1519         print(f'Removing {pdf_path} ...')

```

```

1514     path.unlink()
1515 time_1 = time.time()
1516 print(f'Persisting {pdf_path} ...')
1517 result = abjad.persist(part_file).as_pdf(pdf_path)
1518 print(result[0])
1519 print(result[1])
1520 print(result[2])
1521 success = result[3]
1522 if success is False:
1523     print('LilyPond failed!')
1524 time_2 = time.time()
1525 total_time = time_2 - time_1
1526 print(f'Total time: {total_time} seconds')
1527 if path.exists():
1528     print(f'Opening {pdf_path} ...')
1529     os.system(f'open {pdf_path}')
1530 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/
Invocation.ly').readlines()
1531 open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/Invocation.ly',
'w').writelines(part_lines[15:-1])
1532
1533 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
1534 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1535 signature_copy = abjad.mutate(signatures).copy()
1536 staff_copy = abjad.mutate(staff).copy()
1537 part = abjad.Score()
1538 part.insert(0, staff)
1539 part.insert(0, signature_copy)
1540 part_file = abjad.LilyPondFile.new(
1541     part,
1542     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1543     )
1544 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1'
1545 pdf_path = f'{directory}/Invocation.pdf'
1546 path = pathlib.Path('Invocation.pdf')
1547 if path.exists():
1548     print(f'Removing {pdf_path} ...')
1549     path.unlink()
1550 time_1 = time.time()
1551 print(f'Persisting {pdf_path} ...')
1552 result = abjad.persist(part_file).as_pdf(pdf_path)
1553 print(result[0])
1554 print(result[1])
1555 print(result[2])
1556 success = result[3]
1557 if success is False:
1558     print('LilyPond failed!')
1559 time_2 = time.time()
1560 total_time = time_2 - time_1
1561 print(f'Total time: {total_time} seconds')
1562 if path.exists():
1563     print(f'Opening {pdf_path} ...')
1564     os.system(f'open {pdf_path}')
1565 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/
Invocation.ly').readlines()
1566 open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/Invocation.ly',
'w').writelines(part_lines[15:-1])
1567

```

```

1568 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
1569     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1570     signature_copy = abjad.mutate(signatures).copy()
1571     staff_copy = abjad.mutate(staff).copy()
1572     part = abjad.Score()
1573     part.insert(0, staff)
1574     part.insert(0, signature_copy)
1575     part_file = abjad.LilyPondFile.new(
1576         part,
1577         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
1578         _stylesheets/abjad.ly'],
1579         )
1580     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
1581     pdf_path = f'{directory}/Invocation.pdf'
1582     path = pathlib.Path('Invocation.pdf')
1583     if path.exists():
1584         print(f'Removing {pdf_path} ...')
1585         path.unlink()
1586     time_1 = time.time()
1587     print(f'Persisting {pdf_path} ...')
1588     result = abjad.persist(part_file).as_pdf(pdf_path)
1589     print(result[0])
1590     print(result[1])
1591     print(result[2])
1592     success = result[3]
1593     if success is False:
1594         print('LilyPond failed!')
1595     time_2 = time.time()
1596     total_time = time_2 - time_1
1597     print(f'Total time: {total_time} seconds')
1598     if path.exists():
1599         print(f'Opening {pdf_path} ...')
1600         os.system(f'open {pdf_path}')
1601     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
1602     Invocation.ly').readlines()
1603     open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Invocation.ly', 'w'
1604     ).writelines(part_lines[15:-1])

1605 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
1606     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1607     signature_copy = abjad.mutate(signatures).copy()
1608     staff_copy = abjad.mutate(staff).copy()
1609     part = abjad.Score()
1610     part.insert(0, staff)
1611     part.insert(0, signature_copy)
1612     part_file = abjad.LilyPondFile.new(
1613         part,
1614         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
1615         _stylesheets/abjad.ly'],
1616         )
1617     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
1618     pdf_path = f'{directory}/Invocation.pdf'
1619     path = pathlib.Path('Invocation.pdf')
1620     if path.exists():
1621         print(f'Removing {pdf_path} ...')
1622         path.unlink()
1623     time_1 = time.time()
1624     print(f'Persisting {pdf_path} ...')
1625     result = abjad.persist(part_file).as_pdf(pdf_path)

```

```

1623     print(result[0])
1624     print(result[1])
1625     print(result[2])
1626     success = result[3]
1627     if success is False:
1628         print('LilyPond failed!')
1629     time_2 = time.time()
1630     total_time = time_2 - time_1
1631     print(f'Total time: {total_time} seconds')
1632     if path.exists():
1633         print(f'Opening {pdf_path} ...')
1634         os.system(f'open {pdf_path}')
1635     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.')alto3/
Invocation.ly').readlines()
1636     open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.')alto3/Invocation.ly', 'w
').writelines(part_lines[15:-1])
1637
1638 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
1639     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1640     signature_copy = abjad.mutate(signatures).copy()
1641     staff_copy = abjad.mutate(staff).copy()
1642     part = abjad.Score()
1643     part.insert(0, staff)
1644     part.insert(0, signature_copy)
1645     part_file = abjad.LilyPondFile.new(
1646         part,
1647         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1648         )
1649     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4'
1650     pdf_path = f'{directory}/Invocation.pdf'
1651     path = pathlib.Path('Invocation.pdf')
1652     if path.exists():
1653         print(f'Removing {pdf_path} ...')
1654         path.unlink()
1655     time_1 = time.time()
1656     print(f'Persisting {pdf_path} ...')
1657     result = abjad.persist(part_file).as_pdf(pdf_path)
1658     print(result[0])
1659     print(result[1])
1660     print(result[2])
1661     success = result[3]
1662     if success is False:
1663         print('LilyPond failed!')
1664     time_2 = time.time()
1665     total_time = time_2 - time_1
1666     print(f'Total time: {total_time} seconds')
1667     if path.exists():
1668         print(f'Opening {pdf_path} ...')
1669         os.system(f'open {pdf_path}')
1670     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/
Invocation.ly').readlines()
1671     open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/Invocation.ly', 'w
').writelines(part_lines[15:-1])
1672
1673 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
1674     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1675     signature_copy = abjad.mutate(signatures).copy()
1676     staff_copy = abjad.mutate(staff).copy()

```

```

1677 part = abjad.Score()
1678 part.insert(0, staff)
1679 part.insert(0, signature_copy)
1680 part_file = abjad.LilyPondFile.new(
1681     part,
1682     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1683 _stylesheets/abjad.ily'],
1684     )
1685 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
1686 pdf_path = f'{directory}/Invocation.pdf'
1687 path = pathlib.Path('Invocation.pdf')
1688 if path.exists():
1689     print(f'Removing {pdf_path} ...')
1690     path.unlink()
1691 time_1 = time.time()
1692 print(f'Persisting {pdf_path} ...')
1693 result = abjad.persist(part_file).as_pdf(pdf_path)
1694 print(result[0])
1695 print(result[1])
1696 print(result[2])
1697 success = result[3]
1698 if success is False:
1699     print('LilyPond failed!')
1700 time_2 = time.time()
1701 total_time = time_2 - time_1
1702 print(f'Total time: {total_time} seconds')
1703 if path.exists():
1704     print(f'Opening {pdf_path} ...')
1705     os.system(f'open {pdf_path}')
1706 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/
1707 Invocation.ly').readlines()
1708 open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/Invocation.ly', 'w'
1709 .writelines(part_lines[15:-1])

1710 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
1711     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1712     signature_copy = abjad.mutate(signatures).copy()
1713     staff_copy = abjad.mutate(staff).copy()
1714     part = abjad.Score()
1715     part.insert(0, staff)
1716     part.insert(0, signature_copy)
1717     part_file = abjad.LilyPondFile.new(
1718         part,
1719         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1720 _stylesheets/abjad.ily'],
1721         )
1722     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
1723     pdf_path = f'{directory}/Invocation.pdf'
1724     path = pathlib.Path('Invocation.pdf')
1725     if path.exists():
1726         print(f'Removing {pdf_path} ...')
1727         path.unlink()
1728     time_1 = time.time()
1729     print(f'Persisting {pdf_path} ...')
1730     result = abjad.persist(part_file).as_pdf(pdf_path)
1731     print(result[0])
1732     print(result[1])
1733     print(result[2])
1734     success = result[3]
```

```

1732     if success is False:
1733         print('LilyPond failed!')
1734     time_2 = time.time()
1735     total_time = time_2 - time_1
1736     print(f'Total time: {total_time} seconds')
1737     if path.exists():
1738         print(f'Opening {pdf_path} ...')
1739         os.system(f'open {pdf_path}')
1740     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
1741 Invocation.ly').readlines()
1742     open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Invocation.ly', ,
1743 w').writelines(part_lines[15:-1])
1744
1745 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
1746     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1747     signature_copy = abjad.mutate(signatures).copy()
1748     staff_copy = abjad.mutate(staff).copy()
1749     part = abjad.Score()
1750     part.insert(0, staff)
1751     part.insert(0, signature_copy)
1752     part_file = abjad.LilyPondFile.new(
1753         part,
1754         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1755 _stylesheets/abjad.ily'],
1756         )
1757     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
1758     pdf_path = f'{directory}/Invocation.pdf'
1759     path = pathlib.Path('Invocation.pdf')
1760     if path.exists():
1761         print(f'Removing {pdf_path} ...')
1762         path.unlink()
1763     time_1 = time.time()
1764     print(f'Persisting {pdf_path} ...')
1765     result = abjad.persist(part_file).as_pdf(pdf_path)
1766     print(result[0])
1767     print(result[1])
1768     print(result[2])
1769     success = result[3]
1770     if success is False:
1771         print('LilyPond failed!')
1772     time_2 = time.time()
1773     total_time = time_2 - time_1
1774     print(f'Total time: {total_time} seconds')
1775     if path.exists():
1776         print(f'Opening {pdf_path} ...')
1777         os.system(f'open {pdf_path}')
1778     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
1779 Invocation.ly').readlines()
1780     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Invocation.ly', ,
1781 w').writelines(part_lines[15:-1])
1782
1783 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
1784     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1785     signature_copy = abjad.mutate(signatures).copy()
1786     staff_copy = abjad.mutate(staff).copy()
1787     part = abjad.Score()
1788     part.insert(0, staff)
1789     part.insert(0, signature_copy)
1790     part_file = abjad.LilyPondFile.new(

```

```

1786     part,
1787     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1788 _stylesheets/abjad.ily'],
1789     )
1790 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
1790 pdf_path = f'{directory}/Invocation.pdf'
1791 path = pathlib.Path('Invocation.pdf')
1792 if path.exists():
1793     print(f'Removing {pdf_path} ...')
1794     path.unlink()
1795 time_1 = time.time()
1796 print(f'Persisting {pdf_path} ...')
1797 result = abjad.persist(part_file).as_pdf(pdf_path)
1798 print(result[0])
1799 print(result[1])
1800 print(result[2])
1801 success = result[3]
1802 if success is False:
1803     print('LilyPond failed!')
1804 time_2 = time.time()
1805 total_time = time_2 - time_1
1806 print(f'Total time: {total_time} seconds')
1807 if path.exists():
1808     print(f'Opening {pdf_path} ...')
1809     os.system(f'open {pdf_path}')
1810 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/
1810 Invocation.ly').readlines()
1811 open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/Invocation.ly',
1811 'w').writelines(part_lines[15:-1])
1812
1813 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
1814     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1815     signature_copy = abjad.mutate(signatures).copy()
1816     staff_copy = abjad.mutate(staff).copy()
1817     part = abjad.Score()
1818     part.insert(0, staff)
1819     part.insert(0, signature_copy)
1820     part_file = abjad.LilyPondFile.new(
1821         part,
1822         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1822 _stylesheets/abjad.ily'],
1823         )
1824 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3'
1825 pdf_path = f'{directory}/Invocation.pdf'
1826 path = pathlib.Path('Invocation.pdf')
1827 if path.exists():
1828     print(f'Removing {pdf_path} ...')
1829     path.unlink()
1830 time_1 = time.time()
1831 print(f'Persisting {pdf_path} ...')
1832 result = abjad.persist(part_file).as_pdf(pdf_path)
1833 print(result[0])
1834 print(result[1])
1835 print(result[2])
1836 success = result[3]
1837 if success is False:
1838     print('LilyPond failed!')
1839 time_2 = time.time()
1840 total_time = time_2 - time_1

```

```

1841 print(f'Total time: {total_time} seconds')
1842 if path.exists():
1843     print(f'Opening {pdf_path} ...')
1844     os.system(f'open {pdf_path}')
1845 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
1846 Invocation.ly').readlines()
1847 open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Invocation.ly',
1848 'w').writelines(part_lines[15:-1])
1849
1850 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
1851     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1852     signature_copy = abjad.mutate(signatures).copy()
1853     staff_copy = abjad.mutate(staff).copy()
1854     part = abjad.Score()
1855     part.insert(0, staff)
1856     part.insert(0, signature_copy)
1857     part_file = abjad.LilyPondFile.new(
1858         part,
1859         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1860 _stylesheets/abjad.ily'],
1861         )
1862     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'
1863     pdf_path = f'{directory}/Invocation.pdf'
1864     path = pathlib.Path('Invocation.pdf')
1865     if path.exists():
1866         print(f'Removing {pdf_path} ...')
1867         path.unlink()
1868     time_1 = time.time()
1869     print(f'Persisting {pdf_path} ...')
1870     result = abjad.persist(part_file).as_pdf(pdf_path)
1871     print(result[0])
1872     print(result[1])
1873     print(result[2])
1874     success = result[3]
1875     if success is False:
1876         print('LilyPond failed!')
1877     time_2 = time.time()
1878     total_time = time_2 - time_1
1879     print(f'Total time: {total_time} seconds')
1880     if path.exists():
1881         print(f'Opening {pdf_path} ...')
1882         os.system(f'open {pdf_path}')
1883     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
1884 Invocation.ly').readlines()
1885     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Invocation.ly',
1886 'w').writelines(part_lines[15:-1])
1887
1888 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
1889     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1890     signature_copy = abjad.mutate(signatures).copy()
1891     staff_copy = abjad.mutate(staff).copy()
1892     part = abjad.Score()
1893     part.insert(0, staff)
1894     part.insert(0, signature_copy)
1895     part_file = abjad.LilyPondFile.new(
1896         part,
1897         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1898 _stylesheets/abjad.ily'],
1899         )

```

```

1894 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5'
1895 pdf_path = f'{directory}/Invocation.pdf'
1896 path = pathlib.Path('Invocation.pdf')
1897 if path.exists():
1898     print(f'Removing {pdf_path} ...')
1899     path.unlink()
1900 time_1 = time.time()
1901 print(f'Persisting {pdf_path} ...')
1902 result = abjad.persist(part_file).as_pdf(pdf_path)
1903 print(result[0])
1904 print(result[1])
1905 print(result[2])
1906 success = result[3]
1907 if success is False:
1908     print('LilyPond failed!')
1909 time_2 = time.time()
1910 total_time = time_2 - time_1
1911 print(f'Total time: {total_time} seconds')
1912 if path.exists():
1913     print(f'Opening {pdf_path} ...')
1914     os.system(f'open {pdf_path}')
1915 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/
1916 Invocation.ly').readlines()
1917 open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/Invocation.ly',
1918 'w').writelines(part_lines[15:-1])
1919
1920 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
1921     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1922     signature_copy = abjad.mutate(signatures).copy()
1923     staff_copy = abjad.mutate(staff).copy()
1924     part = abjad.Score()
1925     part.insert(0, staff)
1926     part.insert(0, signature_copy)
1927     part_file = abjad.LilyPondFile.new(
1928         part,
1929         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1930         _stylesheets/abjad.ily'],
1931         )
1932     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1'
1933     pdf_path = f'{directory}/Invocation.pdf'
1934     path = pathlib.Path('Invocation.pdf')
1935     if path.exists():
1936         print(f'Removing {pdf_path} ...')
1937         path.unlink()
1938     time_1 = time.time()
1939     print(f'Persisting {pdf_path} ...')
1940     result = abjad.persist(part_file).as_pdf(pdf_path)
1941     print(result[0])
1942     print(result[1])
1943     print(result[2])
1944     success = result[3]
1945     if success is False:
1946         print('LilyPond failed!')
1947     time_2 = time.time()
1948     total_time = time_2 - time_1
1949     print(f'Total time: {total_time} seconds')
1950     if path.exists():
1951         print(f'Opening {pdf_path} ...')
1952         os.system(f'open {pdf_path}')

```

```

1950 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/
1951 Invocation.ly').readlines()
1952 open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/Invocation.ly
1953 , 'w').writelines(part_lines[15:-1])
1954
1955 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
1956 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1957 signature_copy = abjad.mutate(signatures).copy()
1958 staff_copy = abjad.mutate(staff).copy()
1959 part = abjad.Score()
1960 part.insert(0, staff)
1961 part.insert(0, signature_copy)
1962 part_file = abjad.LilyPondFile.new(
1963     part,
1964     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1965 _stylesheets/abjad.ily'],
1966     )
1967 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2'
1968 pdf_path = f'{directory}/Invocation.pdf'
1969 path = pathlib.Path('Invocation.pdf')
1970 if path.exists():
1971     print(f'Removing {pdf_path} ...')
1972     path.unlink()
1973 time_1 = time.time()
1974 print(f'Persisting {pdf_path} ...')
1975 result = abjad.persist(part_file).as_pdf(pdf_path)
1976 print(result[0])
1977 print(result[1])
1978 print(result[2])
1979 success = result[3]
1980 if success is False:
1981     print('LilyPond failed!')
1982 time_2 = time.time()
1983 total_time = time_2 - time_1
1984 print(f'Total time: {total_time} seconds')
1985 if path.exists():
1986     print(f'Opening {pdf_path} ...')
1987     os.system(f'open {pdf_path}')
1988 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/
1989 Invocation.ly').readlines()
1990 open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/Invocation.ly
1991 , 'w').writelines(part_lines[15:-1])
1992
1993 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
1994 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1995 signature_copy = abjad.mutate(signatures).copy()
1996 staff_copy = abjad.mutate(staff).copy()
1997 part = abjad.Score()
1998 part.insert(0, staff)
1999 part.insert(0, signature_copy)
2000 part_file = abjad.LilyPondFile.new(
2001     part,
2002     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2003 _stylesheets/abjad.ily'],
2004     )
2005 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3'
2006 pdf_path = f'{directory}/Invocation.pdf'
2007 path = pathlib.Path('Invocation.pdf')
2008 if path.exists():

```

```

2003     print(f'Removing {pdf_path} ...')
2004     path.unlink()
2005 time_1 = time.time()
2006 print(f'Persisting {pdf_path} ...')
2007 result = abjad.persist(part_file).as_pdf(pdf_path)
2008 print(result[0])
2009 print(result[1])
2010 print(result[2])
2011 success = result[3]
2012 if success is False:
2013     print('LilyPond failed!')
2014 time_2 = time.time()
2015 total_time = time_2 - time_1
2016 print(f'Total time: {total_time} seconds')
2017 if path.exists():
2018     print(f'Opening {pdf_path} ...')
2019     os.system(f'open {pdf_path}')
2020 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/
Invocation.ly').readlines()
2021 open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/Invocation.ly',
'w').writelines(part_lines[15:-1])
2022
2023 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
2024 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2025 signature_copy = abjad.mutate(signatures).copy()
2026 staff_copy = abjad.mutate(staff).copy()
2027 part = abjad.Score()
2028 part.insert(0, staff)
2029 part.insert(0, signature_copy)
2030 part_file = abjad.LilyPondFile.new(
2031     part,
2032     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2033     )
2034 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
2035 pdf_path = f'{directory}/Invocation.pdf'
2036 path = pathlib.Path('Invocation.pdf')
2037 if path.exists():
2038     print(f'Removing {pdf_path} ...')
2039     path.unlink()
2040 time_1 = time.time()
2041 print(f'Persisting {pdf_path} ...')
2042 result = abjad.persist(part_file).as_pdf(pdf_path)
2043 print(result[0])
2044 print(result[1])
2045 print(result[2])
2046 success = result[3]
2047 if success is False:
2048     print('LilyPond failed!')
2049 time_2 = time.time()
2050 total_time = time_2 - time_1
2051 print(f'Total time: {total_time} seconds')
2052 if path.exists():
2053     print(f'Opening {pdf_path} ...')
2054     os.system(f'open {pdf_path}')
2055 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
Invocation.ly').readlines()
2056 open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Invocation.ly',
'w').writelines(part_lines[15:-1])

```

```

2057 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
2058     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2059     signature_copy = abjad.mutate(signatures).copy()
2060     staff_copy = abjad.mutate(staff).copy()
2061     part = abjad.Score()
2062     part.insert(0, staff)
2063     part.insert(0, signature_copy)
2064     part_file = abjad.LilyPondFile.new(
2065         part,
2066         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2067         _stylesheets/abjad.ily'],
2068         )
2069     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
2070     pdf_path = f'{directory}/Invocation.pdf'
2071     path = pathlib.Path('Invocation.pdf')
2072     if path.exists():
2073         print(f'Removing {pdf_path} ...')
2074         path.unlink()
2075     time_1 = time.time()
2076     print(f'Persisting {pdf_path} ...')
2077     result = abjad.persist(part_file).as_pdf(pdf_path)
2078     print(result[0])
2079     print(result[1])
2080     print(result[2])
2081     success = result[3]
2082     if success is False:
2083         print('LilyPond failed!')
2084     time_2 = time.time()
2085     total_time = time_2 - time_1
2086     print(f'Total time: {total_time} seconds')
2087     if path.exists():
2088         print(f'Opening {pdf_path} ...')
2089         os.system(f'open {pdf_path}')
2090     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
2091     Invocation.ly').readlines()
2092     open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Invocation.ly', 'w').writelines(part_lines[15:-1])
2093 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
2094     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2095     signature_copy = abjad.mutate(signatures).copy()
2096     staff_copy = abjad.mutate(staff).copy()
2097     part = abjad.Score()
2098     part.insert(0, staff)
2099     part.insert(0, signature_copy)
2100     part_file = abjad.LilyPondFile.new(
2101         part,
2102         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2103         _stylesheets/abjad.ily'],
2104         )
2105     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
2106     pdf_path = f'{directory}/Invocation.pdf'
2107     path = pathlib.Path('Invocation.pdf')
2108     if path.exists():
2109         print(f'Removing {pdf_path} ...')
2110         path.unlink()
2111     time_1 = time.time()
2112     print(f'Persisting {pdf_path} ...')

```

```

2112     result = abjad.persist(part_file).as_pdf(pdf_path)
2113     print(result[0])
2114     print(result[1])
2115     print(result[2])
2116     success = result[3]
2117     if success is False:
2118         print('LilyPond failed!')
2119     time_2 = time.time()
2120     total_time = time_2 - time_1
2121     print(f'Total time: {total_time} seconds')
2122     if path.exists():
2123         print(f'Opening {pdf_path} ...')
2124         os.system(f'open {pdf_path}')
2125     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.')contrabass
2126     /Invocation.ly').readlines()
2126     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.')contrabass/Invocation.
2126     ly', 'w').writelines(part_lines[15:-1])

```

Listing 3.1: Invocation Source Code

## 3.2 Section A

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11
12 print('Interpreting file ...')
13
14 time_signatures = [
15     abjad.TimeSignature(pair) for pair in [
16         (5, 4), (4, 4), (3, 4), (4, 4), (3, 4), (4, 4),
17         (5, 4), (5, 4), (4, 4), (3, 4), (4, 4), (3, 4),
18         (4, 4), (5, 4), (5, 4), (4, 4), (3, 4), (4, 4),
19         (3, 4), (4, 4), (4, 4),
20     ]
21 ]
22
23 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
24
25 def cyc(lst):
26     count = 0
27     while True:
28         yield lst[count%len(lst)]
29         count += 1
30
31 def grouper(lst1, lst2):
32     def cyc(lst):
33         c = 0
34         while True:
35             yield lst[c%len(lst)]

```

```

36         c += 1
37     lst1 = cyc(lst1)
38     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2
39 ]
40
40 def reduceMod(list_length, rw):
41     return [(x % list_length) for x in rw]
42
43 sopranino_note = [27, 11, 17, 8, 0, 17, 11, 8, ]
44 soprano_1_note = [22, 5, 16, 13, ]
45 soprano_2_note = [16, 22, 13, 5, ]
46 soprano_3_note = [13, 16, 5, 13, 22, ]
47 alto_1_note = [20, 23, 1, 12, ]
48 alto_2_note = [12, 1, 23, 20, ]
49 alto_3_note = [1, 23, 12, 20, ]
50 alto_4_note = [20, 12, 23, 1, ]
51 alto_5_note = [12, 20, 1, 23, ]
52 alto_6_note = [1, 20, 23, 12, ]
53 tenor_1_note = [17, 25, 6, -1, ]
54 tenor_2_note = [6, -1, 25, 17, ]
55 tenor_3_note = [-1, 6, 25, 17, ]
56 tenor_4_note = [6, 17, 25, -1, ]
57 tenor_5_note = [-1, 17, 6, 25, ]
58 baritone_1_note = [13, 24, 6, 4, ]
59 baritone_2_note = [6, 4, 24, 13, ]
60 baritone_3_note = [4, 6, 13, 24, ]
61 bass_1_note = [11, 18, 9, 0, ]
62 bass_2_note = [9, 11, 0, 18, ]
63 contrabass_note = [-2, 7, 16, 2, 18, 25, ]
64 # -3 at bottom of chord for completion
65 sopranino_chord = [17, 27, 11, 0, 8, ]
66 soprano_1_chord = [[13.25, 16, 26.25, ], ]
67 soprano_2_chord = [[13, 14.75, 26.25, ], ] #maybe it's 13.25?
68 soprano_3_chord = [[12.75, 15.5, 26, ], ]
69 alto_1_chord = [[12.5, 19, 27.75, 34, ], ]
70 alto_2_chord = [[12.5, 15.25, 25.5, ], ]
71 alto_3_chord = [[1.75, 13.5, 22.25, 27, 30, ], ]
72 alto_4_chord = [[12.5, 15.25, 25.5, ], ]
73 alto_5_chord = [[1.75, 13.5, 22.25, 27, 30, ], ]
74 alto_6_chord = [[12.5, 19, 27.75, 34, ], ]
75 tenor_1_chord = [[6, 17.5, ], ]
76 tenor_2_chord = [[6, 17.5, 25.5, 30, ], ]
77 tenor_3_chord = [[6, 17.5, 25.5, 30.75, ], ]
78 tenor_4_chord = [[6, 17.5, ], ]
79 tenor_5_chord = [[6, 17.5, 25.5, 30.75, ], ]
80 baritone_1_chord = [[13.25, 27.5, 33.75, ], ]
81 baritone_2_chord = [[4, 16.5, 23.5, ], ]
82 baritone_3_chord = [[7.75, 17.75, 25.5, 34, ], ]
83 bass_1_chord = [11, 9, 18, ]
84 bass_2_chord = [9, 11, 18, ]
85 contrabass_chord = [-2, 2, 7, -2, 7, 2, ]
86
87 seed(1)
88 sopranino_random_walk = []
89 sopranino_random_walk.append(-1 if random() < 0.5 else 1)
90 for i in range(1, 1000):
91     movement = -1 if random() < 0.5 else 1
92     value = sopranino_random_walk[i-1] + movement
93     sopranino_random_walk.append(value)

```

```
94 sopranino_random_walk_notes = [((x / 2.0) + 19) for x in sopranino_random_walk]
95
96 seed(2)
97 soprano_1_random_walk = []
98 soprano_1_random_walk.append(-1 if random() < 0.5 else 1)
99 for i in range(1, 1000):
100     movement = -1 if random() < 0.5 else 1
101     value = soprano_1_random_walk[i-1] + movement
102     soprano_1_random_walk.append(value)
103 soprano_1_random_walk_notes = [((x / 2.0) + 18) for x in soprano_1_random_walk]
104
105 seed(3)
106 soprano_2_random_walk = []
107 soprano_2_random_walk.append(-1 if random() < 0.5 else 1)
108 for i in range(1, 1000):
109     movement = -1 if random() < 0.5 else 1
110     value = soprano_2_random_walk[i-1] + movement
111     soprano_2_random_walk.append(value)
112 soprano_2_random_walk_notes = [((x / 2.0) + 17) for x in soprano_2_random_walk]
113
114 seed(4)
115 soprano_3_random_walk = []
116 soprano_3_random_walk.append(-1 if random() < 0.5 else 1)
117 for i in range(1, 1000):
118     movement = -1 if random() < 0.5 else 1
119     value = soprano_3_random_walk[i-1] + movement
120     soprano_3_random_walk.append(value)
121 soprano_3_random_walk_notes = [((x / 2.0) + 16) for x in soprano_3_random_walk]
122
123 seed(5)
124 alto_1_random_walk = []
125 alto_1_random_walk.append(-1 if random() < 0.5 else 1)
126 for i in range(1, 1000):
127     movement = -1 if random() < 0.5 else 1
128     value = alto_1_random_walk[i-1] + movement
129     alto_1_random_walk.append(value)
130 alto_1_random_walk_notes = [((x / 2.0) + 14) for x in alto_1_random_walk]
131
132 seed(6)
133 alto_2_random_walk = []
134 alto_2_random_walk.append(-1 if random() < 0.5 else 1)
135 for i in range(1, 1000):
136     movement = -1 if random() < 0.5 else 1
137     value = alto_2_random_walk[i-1] + movement
138     alto_2_random_walk.append(value)
139 alto_2_random_walk_notes = [((x / 2.0) + 13) for x in alto_2_random_walk]
140
141 seed(7)
142 alto_3_random_walk = []
143 alto_3_random_walk.append(-1 if random() < 0.5 else 1)
144 for i in range(1, 1000):
145     movement = -1 if random() < 0.5 else 1
146     value = alto_3_random_walk[i-1] + movement
147     alto_3_random_walk.append(value)
148 alto_3_random_walk_notes = [((x / 2.0) + 12) for x in alto_3_random_walk]
149
150 seed(8)
151 alto_4_random_walk = []
152 alto_4_random_walk.append(-1 if random() < 0.5 else 1)
```

```

153 for i in range(1, 1000):
154     movement = -1 if random() < 0.5 else 1
155     value = alto_4_random_walk[i-1] + movement
156     alto_4_random_walk.append(value)
157 alto_4_random_walk_notes = [((x / 2.0) + 11) for x in alto_4_random_walk]
158
159 seed(9)
160 alto_5_random_walk = []
161 alto_5_random_walk.append(-1 if random() < 0.5 else 1)
162 for i in range(1, 1000):
163     movement = -1 if random() < 0.5 else 1
164     value = alto_5_random_walk[i-1] + movement
165     alto_5_random_walk.append(value)
166 alto_5_random_walk_notes = [((x / 2.0) + 10) for x in alto_5_random_walk]
167
168 seed(10)
169 alto_6_random_walk = []
170 alto_6_random_walk.append(-1 if random() < 0.5 else 1)
171 for i in range(1, 1000):
172     movement = -1 if random() < 0.5 else 1
173     value = alto_6_random_walk[i-1] + movement
174     alto_6_random_walk.append(value)
175 alto_6_random_walk_notes = [((x / 2.0) + 10) for x in alto_6_random_walk]
176
177 seed(11)
178 tenor_1_random_walk = []
179 tenor_1_random_walk.append(-1 if random() < 0.5 else 1)
180 for i in range(1, 1000):
181     movement = -1 if random() < 0.5 else 1
182     value = tenor_1_random_walk[i-1] + movement
183     tenor_1_random_walk.append(value)
184 tenor_1_random_walk_notes = [((x / 2.0) + 9) for x in tenor_1_random_walk]
185
186 seed(12)
187 tenor_2_random_walk = []
188 tenor_2_random_walk.append(-1 if random() < 0.5 else 1)
189 for i in range(1, 1000):
190     movement = -1 if random() < 0.5 else 1
191     value = tenor_2_random_walk[i-1] + movement
192     tenor_2_random_walk.append(value)
193 tenor_2_random_walk_notes = [((x / 2.0) + 8) for x in tenor_2_random_walk]
194
195 seed(13)
196 tenor_3_random_walk = []
197 tenor_3_random_walk.append(-1 if random() < 0.5 else 1)
198 for i in range(1, 1000):
199     movement = -1 if random() < 0.5 else 1
200     value = tenor_3_random_walk[i-1] + movement
201     tenor_3_random_walk.append(value)
202 tenor_3_random_walk_notes = [((x / 2.0) + 7) for x in tenor_3_random_walk]
203
204 seed(14)
205 tenor_4_random_walk = []
206 tenor_4_random_walk.append(-1 if random() < 0.5 else 1)
207 for i in range(1, 1000):
208     movement = -1 if random() < 0.5 else 1
209     value = tenor_4_random_walk[i-1] + movement
210     tenor_4_random_walk.append(value)
211 tenor_4_random_walk_notes = [((x / 2.0) + 6) for x in tenor_4_random_walk]
```

```

212
213 seed(15)
214 tenor_5_random_walk = []
215 tenor_5_random_walk.append(-1 if random() < 0.5 else 1)
216 for i in range(1, 1000):
217     movement = -1 if random() < 0.5 else 1
218     value = tenor_5_random_walk[i-1] + movement
219     tenor_5_random_walk.append(value)
220 tenor_5_random_walk_notes = [((x / 2.0) + 6) for x in tenor_5_random_walk]
221
222 seed(16)
223 baritone_1_random_walk = []
224 baritone_1_random_walk.append(-1 if random() < 0.5 else 1)
225 for i in range(1, 1000):
226     movement = -1 if random() < 0.5 else 1
227     value = baritone_1_random_walk[i-1] + movement
228     baritone_1_random_walk.append(value)
229 baritone_1_random_walk_notes = [((x / 2.0) + 5) for x in baritone_1_random_walk]
230
231 seed(17)
232 baritone_2_random_walk = []
233 baritone_2_random_walk.append(-1 if random() < 0.5 else 1)
234 for i in range(1, 1000):
235     movement = -1 if random() < 0.5 else 1
236     value = baritone_2_random_walk[i-1] + movement
237     baritone_2_random_walk.append(value)
238 baritone_2_random_walk_notes = [((x / 2.0) + 4) for x in baritone_2_random_walk]
239
240 seed(18)
241 baritone_3_random_walk = []
242 baritone_3_random_walk.append(-1 if random() < 0.5 else 1)
243 for i in range(1, 1000):
244     movement = -1 if random() < 0.5 else 1
245     value = baritone_3_random_walk[i-1] + movement
246     baritone_3_random_walk.append(value)
247 baritone_3_random_walk_notes = [((x / 2.0) + 3) for x in baritone_3_random_walk]
248
249 seed(19)
250 bass_1_random_walk = []
251 bass_1_random_walk.append(-1 if random() < 0.5 else 1)
252 for i in range(1, 1000):
253     movement = -1 if random() < 0.5 else 1
254     value = bass_1_random_walk[i-1] + movement
255     bass_1_random_walk.append(value)
256 bass_1_random_walk_notes = [((x / 2.0) + 2) for x in bass_1_random_walk]
257
258 seed(20)
259 bass_2_random_walk = []
260 bass_2_random_walk.append(-1 if random() < 0.5 else 1)
261 for i in range(1, 1000):
262     movement = -1 if random() < 0.5 else 1
263     value = bass_2_random_walk[i-1] + movement
264     bass_2_random_walk.append(value)
265 bass_2_random_walk_notes = [((x / 2.0) + 1) for x in bass_2_random_walk]
266
267 seed(21)
268 contrabass_random_walk = []
269 contrabass_random_walk.append(-1 if random() < 0.5 else 1)
270 for i in range(1, 1000):

```

```

271     movement = -1 if random() < 0.5 else 1
272     value = contrabass_random_walk[i-1] + movement
273     contrabass_random_walk.append(value)
274 contrabass_random_walk_notes = [(x / 2.0) for x in contrabass_random_walk]
275
276 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
277     talea=abjadext.rmakers.Talea(
278         counts=[19, -1, 17, -1, 15, -1, 13, -1, 11, -1, 9, -1, 7, -1, ],
279         denominator=16,
280     ),
281     beamSpecifier=abjadext.rmakers.BeamSpecifier(
282         beamDivisionsTogether=True,
283         beamRests=False,
284     ),
285     extraCountsPerDivision=[0, 1, -1, 0, ],
286     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
287         trivialize=True,
288         extractTrivial=True,
289         rewriteRestFilled=True,
290         rewriteSustained=True,
291     ),
292 )
293
294 rmaker_two = abjadext.rmakers.TaleaRhythmMaker(
295     talea=abjadext.rmakers.Talea(
296         counts=[1, 1, 1, 1, -1, 2, 2, 1, -2, 1, 3, 2, 2, 3, 2, -1, 1, 2, 1, -1, 1,
297         3, ],
298         denominator=16,
299     ),
300     beamSpecifier=abjadext.rmakers.BeamSpecifier(
301         beamDivisionsTogether=True,
302         beamRests=False,
303     ),
304     extraCountsPerDivision=[-1, 0, 1, 0, ],
305     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
306         trivialize=True,
307         extractTrivial=True,
308         rewriteRestFilled=True,
309     ),
310 )
311 attachmentHandler_one = AttachmentHandler(
312     startingDynamic='p',
313     endingDynamic='mp',
314     hairpin='--',
315     articulationList=['tenuto'],
316 )
317 attachmentHandler_two = AttachmentHandler(
318     startingDynamic='mp',
319     endingDynamic='f',
320     hairpin='<',
321     articulationList=['espressivo'],
322 )
323 attachmentHandler_three = AttachmentHandler(
324     startingDynamic='mf',
325     endingDynamic='pp',
326     hairpin='>',
327

```

```

329     articulation_list=['portato', ' ', ' ', ' ', ' ', ' ', ],
330 )
331
332 #####sopranino#####
333 sopranino_musicmaker_one = MusicMaker(
334     rmaker=rmaker_one,
335     pitches=sopranino_note,
336     continuous=True,
337     attachment_handler=attachment_handler_one,
338 )
339 sopranino_musicmaker_two = MusicMaker(
340     rmaker=rmaker_one,
341     pitches=sopranino_chord,
342     continuous=True,
343     attachment_handler=attachment_handler_two,
344 )
345 sopranino_musicmaker_three = MusicMaker(
346     rmaker=rmaker_two,
347     pitches=sopranino_random_walk_notes,
348     continuous=True,
349     attachment_handler=attachment_handler_three,
350 )
351 #####soprano_one#####
352 soprano_one_musicmaker_one = MusicMaker(
353     rmaker=rmaker_one,
354     pitches=soprano_1_note,
355     continuous=True,
356     attachment_handler=attachment_handler_one,
357 )
358 soprano_one_musicmaker_two = MusicMaker(
359     rmaker=rmaker_one,
360     pitches=soprano_1_chord,
361     continuous=True,
362     attachment_handler=attachment_handler_two,
363 )
364 soprano_one_musicmaker_three = MusicMaker(
365     rmaker=rmaker_two,
366     pitches=soprano_1_random_walk_notes,
367     continuous=True,
368     attachment_handler=attachment_handler_three,
369 )
370 #####soprano_two#####
371 soprano_two_musicmaker_one = MusicMaker(
372     rmaker=rmaker_one,
373     pitches=soprano_2_note,
374     continuous=True,
375     attachment_handler=attachment_handler_one,
376 )
377 soprano_two_musicmaker_two = MusicMaker(
378     rmaker=rmaker_one,
379     pitches=soprano_2_chord,
380     continuous=True,
381     attachment_handler=attachment_handler_two,
382 )
383 soprano_two_musicmaker_three = MusicMaker(
384     rmaker=rmaker_two,
385     pitches=soprano_2_random_walk_notes,
386     continuous=True,
387     attachment_handler=attachment_handler_three,

```

```
388 )
389 #####soprano_three#####
390 soprano_three_musicmaker_one = MusicMaker(
391     rmaker=rmaker_one,
392     pitches=soprano_3_note,
393     continuous=True,
394     attachment_handler=attachment_handler_one,
395 )
396 soprano_three_musicmaker_two = MusicMaker(
397     rmaker=rmaker_one,
398     pitches=soprano_3_chord,
399     continuous=True,
400     attachment_handler=attachment_handler_two,
401 )
402 soprano_three_musicmaker_three = MusicMaker(
403     rmaker=rmaker_two,
404     pitches=soprano_3_random_walk_notes,
405     continuous=True,
406     attachment_handler=attachment_handler_three,
407 )
408 #####alto_one#####
409 alto_one_musicmaker_one = MusicMaker(
410     rmaker=rmaker_one,
411     pitches=alto_1_note,
412     continuous=True,
413     attachment_handler=attachment_handler_one,
414 )
415 alto_one_musicmaker_two = MusicMaker(
416     rmaker=rmaker_one,
417     pitches=alto_1_chord,
418     continuous=True,
419     attachment_handler=attachment_handler_two,
420 )
421 alto_one_musicmaker_three = MusicMaker(
422     rmaker=rmaker_two,
423     pitches=soprano_1_random_walk_notes,
424     continuous=True,
425     attachment_handler=attachment_handler_three,
426 )
427 #####alto_two#####
428 alto_two_musicmaker_one = MusicMaker(
429     rmaker=rmaker_one,
430     pitches=alto_2_note,
431     continuous=True,
432     attachment_handler=attachment_handler_one,
433 )
434 alto_two_musicmaker_two = MusicMaker(
435     rmaker=rmaker_one,
436     pitches=alto_2_chord,
437     continuous=True,
438     attachment_handler=attachment_handler_two,
439 )
440 alto_two_musicmaker_three = MusicMaker(
441     rmaker=rmaker_two,
442     pitches=soprano_2_random_walk_notes,
443     continuous=True,
444     attachment_handler=attachment_handler_three,
445 )
446 #####alto_three#####

```

```
447 alto_three_musicmaker_one = MusicMaker(  
448     rmaker=rmaker_one,  
449     pitches=alto_3_note,  
450     continuous=True,  
451     attachment_handler=attachment_handler_one,  
452 )  
453 alto_three_musicmaker_two = MusicMaker(  
454     rmaker=rmaker_one,  
455     pitches=alto_3_chord,  
456     continuous=True,  
457     attachment_handler=attachment_handler_two,  
458 )  
459 alto_three_musicmaker_three = MusicMaker(  
460     rmaker=rmaker_two,  
461     pitches=soprano_3_random_walk_notes,  
462     continuous=True,  
463     attachment_handler=attachment_handler_three,  
464 )  
465 #####alto_four#####  
466 alto_four_musicmaker_one = MusicMaker(  
467     rmaker=rmaker_one,  
468     pitches=alto_4_note,  
469     continuous=True,  
470     attachment_handler=attachment_handler_one,  
471 )  
472 alto_four_musicmaker_two = MusicMaker(  
473     rmaker=rmaker_one,  
474     pitches=alto_4_chord,  
475     continuous=True,  
476     attachment_handler=attachment_handler_two,  
477 )  
478 alto_four_musicmaker_three = MusicMaker(  
479     rmaker=rmaker_two,  
480     pitches=alto_4_random_walk_notes,  
481     continuous=True,  
482     attachment_handler=attachment_handler_three,  
483 )  
484 #####alto_five#####  
485 alto_five_musicmaker_one = MusicMaker(  
486     rmaker=rmaker_one,  
487     pitches=alto_5_note,  
488     continuous=True,  
489     attachment_handler=attachment_handler_one,  
490 )  
491 alto_five_musicmaker_two = MusicMaker(  
492     rmaker=rmaker_one,  
493     pitches=alto_5_chord,  
494     continuous=True,  
495     attachment_handler=attachment_handler_two,  
496 )  
497 alto_five_musicmaker_three = MusicMaker(  
498     rmaker=rmaker_two,  
499     pitches=alto_5_random_walk_notes,  
500     continuous=True,  
501     attachment_handler=attachment_handler_three,  
502 )  
503 #####alto_six#####  
504 alto_six_musicmaker_one = MusicMaker(  
505     rmaker=rmaker_one,
```

```
506     pitches=alto_6_note,
507     continuous=True,
508     attachment_handler=attachment_handler_one,
509 )
510 alto_six_musicmaker_two = MusicMaker(
511     rmaker=rmaker_one,
512     pitches=alto_6_chord,
513     continuous=True,
514     attachment_handler=attachment_handler_two,
515 )
516 alto_six_musicmaker_three = MusicMaker(
517     rmaker=rmaker_two,
518     pitches=alto_6_random_walk_notes,
519     continuous=True,
520     attachment_handler=attachment_handler_three,
521 )
522 #####tenor_one#####
523 tenor_one_musicmaker_one = MusicMaker(
524     rmaker=rmaker_one,
525     pitches=tenor_1_note,
526     continuous=True,
527     attachment_handler=attachment_handler_one,
528 )
529 tenor_one_musicmaker_two = MusicMaker(
530     rmaker=rmaker_one,
531     pitches=tenor_1_chord,
532     continuous=True,
533     attachment_handler=attachment_handler_two,
534 )
535 tenor_one_musicmaker_three = MusicMaker(
536     rmaker=rmaker_two,
537     pitches=tenor_1_random_walk_notes,
538     continuous=True,
539     attachment_handler=attachment_handler_three,
540 )
541 #####tenor_two#####
542 tenor_two_musicmaker_one = MusicMaker(
543     rmaker=rmaker_one,
544     pitches=tenor_2_note,
545     continuous=True,
546     attachment_handler=attachment_handler_one,
547 )
548 tenor_two_musicmaker_two = MusicMaker(
549     rmaker=rmaker_one,
550     pitches=tenor_2_chord,
551     continuous=True,
552     attachment_handler=attachment_handler_two,
553 )
554 tenor_two_musicmaker_three = MusicMaker(
555     rmaker=rmaker_two,
556     pitches=tenor_2_random_walk_notes,
557     continuous=True,
558     attachment_handler=attachment_handler_three,
559 )
560 #####tenor_three#####
561 tenor_three_musicmaker_one = MusicMaker(
562     rmaker=rmaker_one,
563     pitches=tenor_3_note,
564     continuous=True,
```

```

565     attachment_handler=attachment_handler_one ,
566 )
567 tenor_three_musicmaker_two = MusicMaker(
568     rmaker=rmaker_one ,
569     pitches=tenor_3_chord ,
570     continuous=True ,
571     attachment_handler=attachment_handler_two ,
572 )
573 tenor_three_musicmaker_three = MusicMaker(
574     rmaker=rmaker_two ,
575     pitches=tenor_3_random_walk_notes ,
576     continuous=True ,
577     attachment_handler=attachment_handler_three ,
578 )
579 #####tenor_four#####
580 tenor_four_musicmaker_one = MusicMaker(
581     rmaker=rmaker_one ,
582     pitches=tenor_4_note ,
583     continuous=True ,
584     attachment_handler=attachment_handler_one ,
585 )
586 tenor_four_musicmaker_two = MusicMaker(
587     rmaker=rmaker_one ,
588     pitches=tenor_4_chord ,
589     continuous=True ,
590     attachment_handler=attachment_handler_two ,
591 )
592 tenor_four_musicmaker_three = MusicMaker(
593     rmaker=rmaker_two ,
594     pitches=tenor_4_random_walk_notes ,
595     continuous=True ,
596     attachment_handler=attachment_handler_three ,
597 )
598 #####tenor_five#####
599 tenor_five_musicmaker_one = MusicMaker(
600     rmaker=rmaker_one ,
601     pitches=tenor_5_note ,
602     continuous=True ,
603     attachment_handler=attachment_handler_one ,
604 )
605 tenor_five_musicmaker_two = MusicMaker(
606     rmaker=rmaker_one ,
607     pitches=tenor_5_chord ,
608     continuous=True ,
609     attachment_handler=attachment_handler_two ,
610 )
611 tenor_five_musicmaker_three = MusicMaker(
612     rmaker=rmaker_two ,
613     pitches=tenor_5_random_walk_notes ,
614     continuous=True ,
615     attachment_handler=attachment_handler_three ,
616 )
617 #####baritone_one#####
618 baritone_one_musicmaker_one = MusicMaker(
619     rmaker=rmaker_one ,
620     pitches=baritone_1_note ,
621     continuous=True ,
622     attachment_handler=attachment_handler_one ,
623 )

```

```
624 baritone_one_musicmaker_two = MusicMaker(            
625     rmaker=rmaker_one,                              
626     pitches=baritone_1_chord,                        
627     continuous=True,                              
628     attachment_handler=attachment_handler_two,      
629 )  
630 baritone_one_musicmaker_three = MusicMaker(            
631     rmaker=rmaker_two,                              
632     pitches=baritone_1_random_walk_notes,            
633     continuous=True,                              
634     attachment_handler=attachment_handler_three,    
635 )  
636 #####baritone_two#####  
637 baritone_two_musicmaker_one = MusicMaker(            
638     rmaker=rmaker_one,                              
639     pitches=baritone_2_note,                        
640     continuous=True,                              
641     attachment_handler=attachment_handler_one,      
642 )  
643 baritone_two_musicmaker_two = MusicMaker(            
644     rmaker=rmaker_one,                              
645     pitches=baritone_2_chord,                        
646     continuous=True,                              
647     attachment_handler=attachment_handler_two,      
648 )  
649 baritone_two_musicmaker_three = MusicMaker(            
650     rmaker=rmaker_two,                              
651     pitches=baritone_2_random_walk_notes,            
652     continuous=True,                              
653     attachment_handler=attachment_handler_three,    
654 )  
655 #####baritone_three#####  
656 baritone_three_musicmaker_one = MusicMaker(            
657     rmaker=rmaker_one,                              
658     pitches=baritone_3_note,                        
659     continuous=True,                              
660     attachment_handler=attachment_handler_one,      
661 )  
662 baritone_three_musicmaker_two = MusicMaker(            
663     rmaker=rmaker_one,                              
664     pitches=baritone_3_chord,                        
665     continuous=True,                              
666     attachment_handler=attachment_handler_two,      
667 )  
668 baritone_three_musicmaker_three = MusicMaker(            
669     rmaker=rmaker_two,                              
670     pitches=baritone_3_random_walk_notes,            
671     continuous=True,                              
672     attachment_handler=attachment_handler_three,    
673 )  
674 #####bass_one#####  
675 bass_one_musicmaker_one = MusicMaker(            
676     rmaker=rmaker_one,                              
677     pitches=bass_1_note,                            
678     continuous=True,                              
679     attachment_handler=attachment_handler_one,      
680 )  
681 bass_one_musicmaker_two = MusicMaker(            
682     rmaker=rmaker_one,
```

```
683     pitches=bass_1_chord ,
684     continuous=True ,
685     attachment_handler=attachment_handler_two ,
686 )
687 bass_one_musicmaker_three = MusicMaker(
688     rmaker=rmaker_two ,
689     pitches=bass_1_random_walk_notes ,
690     continuous=True ,
691     attachment_handler=attachment_handler_three ,
692 )
693 #####bass_two#####
694 bass_two_musicmaker_one = MusicMaker(
695     rmaker=rmaker_one ,
696     pitches=bass_2_note ,
697     continuous=True ,
698     attachment_handler=attachment_handler_one ,
699 )
700 bass_two_musicmaker_two = MusicMaker(
701     rmaker=rmaker_one ,
702     pitches=bass_2_chord ,
703     continuous=True ,
704     attachment_handler=attachment_handler_two ,
705 )
706 bass_two_musicmaker_three = MusicMaker(
707     rmaker=rmaker_two ,
708     pitches=bass_2_random_walk_notes ,
709     continuous=True ,
710     attachment_handler=attachment_handler_three ,
711 )
712 #####contrabass#####
713 contrabass_musicmaker_one = MusicMaker(
714     rmaker=rmaker_one ,
715     pitches=contrabass_note ,
716     continuous=True ,
717     attachment_handler=attachment_handler_one ,
718 )
719 contrabass_musicmaker_two = MusicMaker(
720     rmaker=rmaker_one ,
721     pitches=contrabass_chord ,
722     continuous=True ,
723     attachment_handler=attachment_handler_two ,
724 )
725 contrabass_musicmaker_three = MusicMaker(
726     rmaker=rmaker_two ,
727     pitches=contrabass_random_walk_notes ,
728     continuous=True ,
729     attachment_handler=attachment_handler_three ,
730 )
731 silence_maker = abjadext.rmakers.NoteRhythmMaker(
732     division_masks=[
733         abjadext.rmakers.SilenceMask(
734             pattern=abjad.index([0], 1),
735             ),
736         ],
737     ),
738 )
739
740 class MusicSpecifier:
```

```

742     def __init__(self, music_maker, voice_name):
743         self.music_maker = music_maker
744         self.voice_name = voice_name
745
746     print('Collecting timespans and rmakers ...')
747
748     voice_1_timespan_list = abjad.TimespanList([
749         abjad.AnnotatedTimespan(
750             start_offset=start_offset,
751             stop_offset=stop_offset,
752             annotation=MusicSpecifier(
753                 music_maker=music_maker,
754                 voice_name='Voice 1',
755             ),
756         )
757         for start_offset, stop_offset, music_maker in [
758             [(0, 4), (1, 4), sopranino_musicmaker_one],
759             [(1, 4), (2, 4), sopranino_musicmaker_one],
760             [(2, 4), (3, 4), sopranino_musicmaker_one],
761             [(3, 4), (4, 4), sopranino_musicmaker_one],
762             [(4, 4), (5, 4), sopranino_musicmaker_one],
763             [(11, 8), (13, 8), sopranino_musicmaker_one],
764             [(21, 8), (23, 8), sopranino_musicmaker_two],
765             [(12, 4), (13, 4), sopranino_musicmaker_one],
766             [(13, 4), (14, 4), sopranino_musicmaker_one],
767             [(14, 4), (15, 4), sopranino_musicmaker_one],
768             [(35, 8), (37, 8), sopranino_musicmaker_two],
769             [(37, 8), (39, 8), sopranino_musicmaker_two],
770             [(21, 4), (22, 4), sopranino_musicmaker_one],
771             [(22, 4), (23, 4), sopranino_musicmaker_one],
772             [(24, 4), (25, 4), sopranino_musicmaker_two],
773             [(25, 4), (26, 4), sopranino_musicmaker_two],
774             [(26, 4), (27, 4), sopranino_musicmaker_two],
775             [(27, 4), (28, 4), sopranino_musicmaker_two],
776
777             [(28, 4), (29, 4), sopranino_musicmaker_one],
778             [(29, 4), (30, 4), sopranino_musicmaker_one],
779             [(30, 4), (31, 4), sopranino_musicmaker_one],
780             [(31, 4), (32, 4), sopranino_musicmaker_one],
781             [(32, 4), (33, 4), sopranino_musicmaker_one],
782             [(67, 8), (69, 8), sopranino_musicmaker_two],
783             [(39, 4), (40, 4), sopranino_musicmaker_three],
784             [(40, 4), (41, 4), sopranino_musicmaker_one],
785             [(41, 4), (42, 4), sopranino_musicmaker_one],
786             [(42, 4), (43, 4), sopranino_musicmaker_one],
787             [(46, 4), (47, 4), sopranino_musicmaker_two],
788             [(47, 4), (48, 4), sopranino_musicmaker_three],
789             [(49, 4), (50, 4), sopranino_musicmaker_one],
790             [(50, 4), (51, 4), sopranino_musicmaker_one],
791             [(52, 4), (53, 4), sopranino_musicmaker_two],
792             [(53, 4), (54, 4), sopranino_musicmaker_two],
793             [(54, 4), (55, 4), sopranino_musicmaker_two],
794             [(55, 4), (56, 4), sopranino_musicmaker_two],
795
796             [(56, 4), (57, 4), sopranino_musicmaker_one],
797             [(57, 4), (58, 4), sopranino_musicmaker_one],
798             [(58, 4), (59, 4), sopranino_musicmaker_one],
799             [(59, 4), (60, 4), sopranino_musicmaker_one],
800             [(60, 4), (61, 4), sopranino_musicmaker_one],

```

```

801     [(61, 4), (62, 4), soprano_one_musicmaker_one],
802     [(62, 4), (63, 4), soprano_one_musicmaker_one],
803     [(63, 4), (64, 4), soprano_one_musicmaker_one],
804     [(67, 4), (68, 4), soprano_one_musicmaker_two],
805     [(68, 4), (69, 4), soprano_one_musicmaker_two],
806     [(69, 4), (70, 4), soprano_one_musicmaker_two],
807     [(70, 4), (71, 4), soprano_one_musicmaker_two],
808     [(74, 4), (75, 4), soprano_one_musicmaker_three],
809     [(75, 4), (76, 4), soprano_one_musicmaker_three],
810     [(76, 4), (77, 4), soprano_one_musicmaker_three],
811     [(77, 4), (78, 4), soprano_one_musicmaker_one],
812     [(78, 4), (79, 4), soprano_one_musicmaker_one],
813     [(79, 4), (80, 4), soprano_one_musicmaker_two],
814     [(80, 4), (81, 4), soprano_one_musicmaker_two],
815     [(81, 4), (82, 4), soprano_one_musicmaker_two],
816     [(82, 4), (83, 4), soprano_one_musicmaker_two],
817 ]
818 ])
819
820 voice_2_timespan_list = abjad.TimespanList([
821     abjad.AnnotatedTimespan(
822         start_offset=start_offset,
823         stop_offset=stop_offset,
824         annotation=MusicSpecifier(
825             music_maker=music_maker,
826             voice_name='Voice 2',
827         ),
828     )
829     for start_offset, stop_offset, music_maker in [
830         [(0, 4), (1, 4), soprano_one_musicmaker_one],
831         [(1, 4), (2, 4), soprano_one_musicmaker_one],
832         [(2, 4), (3, 4), soprano_one_musicmaker_one],
833         [(3, 4), (4, 4), soprano_one_musicmaker_one],
834         [(4, 4), (5, 4), soprano_one_musicmaker_one],
835         [(9, 4), (10, 4), soprano_one_musicmaker_two],
836         [(10, 4), (11, 4), soprano_one_musicmaker_two],
837         [(11, 4), (12, 4), soprano_one_musicmaker_two],
838         [(14, 4), (15, 4), soprano_one_musicmaker_one],
839         [(21, 4), (22, 4), soprano_one_musicmaker_two],
840         [(22, 4), (23, 4), soprano_one_musicmaker_two],
841         [(23, 4), (24, 4), soprano_one_musicmaker_two],
842         [(24, 4), (25, 4), soprano_one_musicmaker_two],
843         [(25, 4), (26, 4), soprano_one_musicmaker_two],
844
845         [(28, 4), (29, 4), soprano_one_musicmaker_one],
846         [(29, 4), (30, 4), soprano_one_musicmaker_one],
847         [(30, 4), (31, 4), soprano_one_musicmaker_one],
848         [(31, 4), (32, 4), soprano_one_musicmaker_one],
849         [(32, 4), (33, 4), soprano_one_musicmaker_one],
850         [(37, 4), (38, 4), soprano_one_musicmaker_two],
851         [(38, 4), (39, 4), soprano_one_musicmaker_two],
852         [(39, 4), (40, 4), soprano_one_musicmaker_two],
853         [(42, 4), (43, 4), soprano_one_musicmaker_three],
854         [(49, 4), (50, 4), soprano_one_musicmaker_one],
855         [(50, 4), (51, 4), soprano_one_musicmaker_one],
856         [(51, 4), (52, 4), soprano_one_musicmaker_two],
857         [(52, 4), (53, 4), soprano_one_musicmaker_two],
858         [(53, 4), (54, 4), soprano_one_musicmaker_two],
859

```

```

860     [(56, 4), (57, 4), soprano_one_musicmaker_one],
861     [(57, 4), (58, 4), soprano_one_musicmaker_one],
862     [(58, 4), (59, 4), soprano_one_musicmaker_one],
863     [(59, 4), (60, 4), soprano_one_musicmaker_one],
864     [(60, 4), (61, 4), soprano_one_musicmaker_one],
865     [(65, 4), (66, 4), soprano_one_musicmaker_one],
866     [(66, 4), (67, 4), soprano_one_musicmaker_one],
867     [(67, 4), (68, 4), soprano_one_musicmaker_one],
868     [(70, 4), (71, 4), soprano_one_musicmaker_two],
869     [(77, 4), (78, 4), soprano_one_musicmaker_two],
870     [(78, 4), (79, 4), soprano_one_musicmaker_two],
871     [(79, 4), (80, 4), soprano_one_musicmaker_three],
872     [(80, 4), (81, 4), soprano_one_musicmaker_three],
873     [(81, 4), (82, 4), soprano_one_musicmaker_three],
874 ]
875 ])
876
877 voice_3_timestrap_list = abjad.TimespanList([
878     abjad.AnnotatedTimespan(
879         start_offset=start_offset,
880         stop_offset=stop_offset,
881         annotation=MusicSpecifier(
882             music_maker=music_maker,
883             voice_name='Voice 3',
884         ),
885     )
886     for start_offset, stop_offset, music_maker in [
887         [(0, 4), (1, 4), soprano_two_musicmaker_two],
888         [(1, 4), (2, 4), soprano_two_musicmaker_two],
889         [(2, 4), (3, 4), soprano_two_musicmaker_two],
890         [(3, 4), (4, 4), soprano_two_musicmaker_two],
891         [(7, 4), (8, 4), soprano_two_musicmaker_one],
892         [(8, 4), (9, 4), soprano_two_musicmaker_one],
893         [(10, 4), (11, 4), soprano_two_musicmaker_two],
894         [(11, 4), (12, 4), soprano_two_musicmaker_two],
895         [(12, 4), (13, 4), soprano_two_musicmaker_two],
896         [(13, 4), (14, 4), soprano_two_musicmaker_two],
897         [(14, 4), (15, 4), soprano_two_musicmaker_two],
898         [(15, 4), (16, 4), soprano_two_musicmaker_two],
899         [(21, 4), (22, 4), soprano_two_musicmaker_one],
900         [(22, 4), (23, 4), soprano_two_musicmaker_one],
901         [(23, 4), (24, 4), soprano_two_musicmaker_two],
902         [(24, 4), (25, 4), soprano_two_musicmaker_two],
903
904         [(28, 4), (29, 4), soprano_two_musicmaker_one],
905         [(29, 4), (30, 4), soprano_two_musicmaker_one],
906         [(30, 4), (31, 4), soprano_two_musicmaker_one],
907         [(31, 4), (32, 4), soprano_two_musicmaker_one],
908         [(35, 4), (36, 4), soprano_two_musicmaker_two],
909         [(36, 4), (37, 4), soprano_two_musicmaker_two],
910         [(38, 4), (39, 4), soprano_two_musicmaker_three],
911         [(39, 4), (40, 4), soprano_two_musicmaker_three],
912         [(40, 4), (41, 4), soprano_two_musicmaker_one],
913         [(41, 4), (42, 4), soprano_two_musicmaker_one],
914         [(42, 4), (43, 4), soprano_two_musicmaker_one],
915         [(43, 4), (44, 4), soprano_two_musicmaker_one],
916         [(49, 4), (50, 4), soprano_two_musicmaker_two],
917         [(50, 4), (51, 4), soprano_two_musicmaker_two],
918         [(51, 4), (52, 4), soprano_two_musicmaker_three],

```

```

919     [(52, 4), (53, 4), soprano_two_musicmaker_three],
920
921     [(56, 4), (57, 4), soprano_two_musicmaker_one],
922     [(57, 4), (58, 4), soprano_two_musicmaker_one],
923     [(58, 4), (59, 4), soprano_two_musicmaker_one],
924     [(59, 4), (60, 4), soprano_two_musicmaker_one],
925     [(63, 4), (64, 4), soprano_two_musicmaker_one],
926     [(64, 4), (65, 4), soprano_two_musicmaker_one],
927     [(66, 4), (67, 4), soprano_two_musicmaker_two],
928     [(67, 4), (68, 4), soprano_two_musicmaker_two],
929     [(68, 4), (69, 4), soprano_two_musicmaker_two],
930     [(69, 4), (70, 4), soprano_two_musicmaker_two],
931     [(70, 4), (71, 4), soprano_two_musicmaker_two],
932     [(71, 4), (72, 4), soprano_two_musicmaker_two],
933     [(77, 4), (78, 4), soprano_two_musicmaker_three],
934     [(78, 4), (79, 4), soprano_two_musicmaker_three],
935     [(79, 4), (80, 4), soprano_two_musicmaker_three],
936     [(80, 4), (81, 4), soprano_two_musicmaker_three],
937 ]
938 ])
939
940 voice_4_timespan_list = abjad.TimespanList([
941     abjad.AnnotatedTimespan(
942         start_offset=start_offset,
943         stop_offset=stop_offset,
944         annotation=MusicSpecifier(
945             music_maker=music_maker,
946             voice_name='Voice 4',
947         ),
948     )
949     for start_offset, stop_offset, music_maker in [
950         [(0, 4), (1, 4), soprano_three_musicmaker_one],
951         [(1, 4), (2, 4), soprano_three_musicmaker_one],
952         [(2, 4), (3, 4), soprano_three_musicmaker_one],
953         [(5, 4), (6, 4), soprano_three_musicmaker_two],
954         [(12, 4), (13, 4), soprano_three_musicmaker_two],
955         [(13, 4), (14, 4), soprano_three_musicmaker_two],
956         [(14, 4), (15, 4), soprano_three_musicmaker_two],
957         [(15, 4), (16, 4), soprano_three_musicmaker_two],
958         [(16, 4), (17, 4), soprano_three_musicmaker_one],
959         [(21, 4), (22, 4), soprano_three_musicmaker_two],
960         [(22, 4), (23, 4), soprano_three_musicmaker_two],
961         [(23, 4), (24, 4), soprano_three_musicmaker_one],
962         [(26, 4), (27, 4), soprano_three_musicmaker_one],
963
964         [(28, 4), (29, 4), soprano_three_musicmaker_one],
965         [(29, 4), (30, 4), soprano_three_musicmaker_one],
966         [(30, 4), (31, 4), soprano_three_musicmaker_one],
967         [(33, 4), (34, 4), soprano_three_musicmaker_two],
968         [(40, 4), (41, 4), soprano_three_musicmaker_three],
969         [(41, 4), (42, 4), soprano_three_musicmaker_three],
970         [(42, 4), (43, 4), soprano_three_musicmaker_three],
971         [(43, 4), (44, 4), soprano_three_musicmaker_three],
972         [(44, 4), (45, 4), soprano_three_musicmaker_one],
973         [(49, 4), (50, 4), soprano_three_musicmaker_two],
974         [(50, 4), (51, 4), soprano_three_musicmaker_two],
975         [(51, 4), (52, 4), soprano_three_musicmaker_three],
976         [(54, 4), (55, 4), soprano_three_musicmaker_one],
977

```

```

978     [(56, 4), (57, 4), soprano_three_musicmaker_one],
979     [(57, 4), (58, 4), soprano_three_musicmaker_one],
980     [(58, 4), (59, 4), soprano_three_musicmaker_one],
981     [(61, 4), (62, 4), soprano_three_musicmaker_one],
982     [(68, 4), (69, 4), soprano_three_musicmaker_two],
983     [(69, 4), (70, 4), soprano_three_musicmaker_two],
984     [(70, 4), (71, 4), soprano_three_musicmaker_two],
985     [(71, 4), (72, 4), soprano_three_musicmaker_two],
986     [(72, 4), (73, 4), soprano_three_musicmaker_two],
987     [(77, 4), (78, 4), soprano_three_musicmaker_three],
988     [(78, 4), (79, 4), soprano_three_musicmaker_three],
989     [(79, 4), (80, 4), soprano_three_musicmaker_three],
990     [(82, 4), (83, 4), soprano_three_musicmaker_one],
991 ]
992 ])
993
994 voice_5_timestspan_list = abjad.TimespanList([
995     abjad.AnnotatedTimespan(
996         start_offset=start_offset,
997         stop_offset=stop_offset,
998         annotation=MusicSpecifier(
999             music_maker=music_maker,
1000             voice_name='Voice 5',
1001         ),
1002     )
1003     for start_offset, stop_offset, music_maker in [
1004         [(0, 4), (1, 4), alto_one_musicmaker_two],
1005         [(1, 4), (2, 4), alto_one_musicmaker_two],
1006         [(3, 4), (4, 4), alto_one_musicmaker_two],
1007         [(4, 4), (5, 4), alto_one_musicmaker_two],
1008         [(5, 4), (6, 4), alto_one_musicmaker_one],
1009         [(6, 4), (7, 4), alto_one_musicmaker_one],
1010         [(7, 4), (8, 4), alto_one_musicmaker_one],
1011         [(8, 4), (9, 4), alto_one_musicmaker_one],
1012         [(14, 4), (15, 4), alto_one_musicmaker_two],
1013         [(15, 4), (16, 4), alto_one_musicmaker_two],
1014         [(16, 4), (17, 4), alto_one_musicmaker_one],
1015         [(17, 4), (18, 4), alto_one_musicmaker_one],
1016         [(21, 4), (22, 4), alto_one_musicmaker_one],
1017         [(22, 4), (23, 4), alto_one_musicmaker_one],
1018         [(47, 8), (24, 4), alto_one_musicmaker_two],
1019         [(24, 4), (25, 4), alto_one_musicmaker_two],
1020         [(25, 4), (26, 4), alto_one_musicmaker_two],
1021         [(26, 4), (27, 4), alto_one_musicmaker_two],
1022         [(27, 4), (28, 4), alto_one_musicmaker_two],
1023
1024         [(28, 4), (29, 4), alto_one_musicmaker_one],
1025         [(29, 4), (30, 4), alto_one_musicmaker_one],
1026         [(31, 4), (32, 4), alto_one_musicmaker_two],
1027         [(32, 4), (33, 4), alto_one_musicmaker_two],
1028         [(33, 4), (34, 4), alto_one_musicmaker_three],
1029         [(34, 4), (35, 4), alto_one_musicmaker_three],
1030         [(35, 4), (36, 4), alto_one_musicmaker_three],
1031         [(36, 4), (37, 4), alto_one_musicmaker_three],
1032         [(42, 4), (43, 4), alto_one_musicmaker_one],
1033         [(43, 4), (44, 4), alto_one_musicmaker_one],
1034         [(44, 4), (45, 4), alto_one_musicmaker_two],
1035         [(45, 4), (46, 4), alto_one_musicmaker_two],
1036         [(48, 4), (49, 4), alto_one_musicmaker_three],

```

```

1037     [(49, 4), (50, 4), alto_one_musicmaker_three],
1038     [(52, 4), (53, 4), alto_one_musicmaker_one],
1039     [(53, 4), (54, 4), alto_one_musicmaker_one],
1040     [(54, 4), (55, 4), alto_one_musicmaker_one],
1041     [(55, 4), (56, 4), alto_one_musicmaker_one],
1042
1043     [(56, 4), (57, 4), alto_one_musicmaker_one],
1044     [(57, 4), (58, 4), alto_one_musicmaker_one],
1045     [(59, 4), (60, 4), alto_one_musicmaker_one],
1046     [(60, 4), (61, 4), alto_one_musicmaker_one],
1047     [(61, 4), (62, 4), alto_one_musicmaker_two],
1048     [(62, 4), (63, 4), alto_one_musicmaker_two],
1049     [(63, 4), (64, 4), alto_one_musicmaker_two],
1050     [(64, 4), (65, 4), alto_one_musicmaker_two],
1051     [(70, 4), (71, 4), alto_one_musicmaker_two],
1052     [(71, 4), (72, 4), alto_one_musicmaker_two],
1053     [(72, 4), (73, 4), alto_one_musicmaker_three],
1054     [(73, 4), (74, 4), alto_one_musicmaker_three],
1055     [(76, 4), (77, 4), alto_one_musicmaker_three],
1056     [(77, 4), (78, 4), alto_one_musicmaker_three],
1057     [(80, 4), (81, 4), alto_one_musicmaker_one],
1058     [(81, 4), (82, 4), alto_one_musicmaker_one],
1059     [(82, 4), (83, 4), alto_one_musicmaker_one],
1060 ]
1061 ])
1062
1063 voice_6_timespan_list = abjad.TimespanList([
1064     abjad.AnnotatedTimespan(
1065         start_offset=start_offset,
1066         stop_offset=stop_offset,
1067         annotation=MusicSpecifier(
1068             music_maker=music_maker,
1069             voice_name='Voice 6',
1070         ),
1071     ),
1072     for start_offset, stop_offset, music_maker in [
1073         [(0, 4), (1, 4), alto_two_musicmaker_two],
1074         [(7, 4), (8, 4), alto_two_musicmaker_one],
1075         [(8, 4), (9, 4), alto_two_musicmaker_one],
1076         [(9, 4), (10, 4), alto_two_musicmaker_two],
1077         [(10, 4), (11, 4), alto_two_musicmaker_two],
1078         [(11, 4), (12, 4), alto_two_musicmaker_two],
1079         [(16, 4), (17, 4), alto_two_musicmaker_one],
1080         [(17, 4), (18, 4), alto_two_musicmaker_one],
1081         [(18, 4), (19, 4), alto_two_musicmaker_one],
1082         [(21, 4), (22, 4), alto_two_musicmaker_one],
1083
1084         [(28, 4), (29, 4), alto_two_musicmaker_one],
1085         [(35, 4), (36, 4), alto_two_musicmaker_two],
1086         [(36, 4), (37, 4), alto_two_musicmaker_two],
1087         [(37, 4), (38, 4), alto_two_musicmaker_three],
1088         [(38, 4), (39, 4), alto_two_musicmaker_three],
1089         [(39, 4), (40, 4), alto_two_musicmaker_three],
1090         [(44, 4), (45, 4), alto_two_musicmaker_one],
1091         [(45, 4), (46, 4), alto_two_musicmaker_one],
1092         [(46, 4), (47, 4), alto_two_musicmaker_one],
1093         [(49, 4), (50, 4), alto_two_musicmaker_two],
1094
1095         [(56, 4), (57, 4), alto_two_musicmaker_one],

```

```

1096     [(63, 4), (64, 4), alto_two_musicmaker_one],
1097     [(64, 4), (65, 4), alto_two_musicmaker_one],
1098     [(65, 4), (66, 4), alto_two_musicmaker_two],
1099     [(66, 4), (67, 4), alto_two_musicmaker_two],
1100     [(67, 4), (68, 4), alto_two_musicmaker_two],
1101     [(72, 4), (73, 4), alto_two_musicmaker_two],
1102     [(73, 4), (74, 4), alto_two_musicmaker_two],
1103     [(74, 4), (75, 4), alto_two_musicmaker_two],
1104     [(77, 4), (78, 4), alto_two_musicmaker_three],
1105   ],
1106 ]
1107 ]
1108 voice_7_timespan_list = abjad.TimespanList([
1109     abjad.AnnotatedTimespan(
1110         start_offset=start_offset,
1111         stop_offset=stop_offset,
1112         annotation=MusicSpecifier(
1113             music_maker=music_maker,
1114             voice_name='Voice 7',
1115         ),
1116     ),
1117     for start_offset, stop_offset, music_maker in [
1118         [(0, 4), (1, 4), alto_three_musicmaker_one],
1119         [(1, 4), (2, 4), alto_three_musicmaker_one],
1120         [(2, 4), (3, 4), alto_three_musicmaker_one],
1121         [(3, 4), (4, 4), alto_three_musicmaker_one],
1122         [(4, 4), (5, 4), alto_three_musicmaker_one],
1123         [(5, 4), (6, 4), alto_three_musicmaker_two],
1124         [(11, 4), (12, 4), alto_three_musicmaker_one],
1125         [(12, 4), (13, 4), alto_three_musicmaker_one],
1126         [(13, 4), (14, 4), alto_three_musicmaker_one],
1127         [(14, 4), (15, 4), alto_three_musicmaker_one],
1128         [(18, 4), (19, 4), alto_three_musicmaker_two],
1129         [(19, 4), (20, 4), alto_three_musicmaker_one],
1130         [(21, 4), (22, 4), alto_three_musicmaker_two],
1131         [(22, 4), (23, 4), alto_three_musicmaker_two],
1132         [(23, 4), (24, 4), alto_three_musicmaker_two],
1133         [(24, 4), (25, 4), alto_three_musicmaker_two],
1134         [(25, 4), (26, 4), alto_three_musicmaker_two],
1135         [(26, 4), (27, 4), alto_three_musicmaker_two],
1136
1137         [(28, 4), (29, 4), alto_three_musicmaker_one],
1138         [(29, 4), (30, 4), alto_three_musicmaker_one],
1139         [(30, 4), (31, 4), alto_three_musicmaker_one],
1140         [(31, 4), (32, 4), alto_three_musicmaker_one],
1141         [(32, 4), (33, 4), alto_three_musicmaker_one],
1142         [(33, 4), (34, 4), alto_three_musicmaker_two],
1143         [(39, 4), (40, 4), alto_three_musicmaker_three],
1144         [(40, 4), (41, 4), alto_three_musicmaker_one],
1145         [(41, 4), (42, 4), alto_three_musicmaker_one],
1146         [(42, 4), (43, 4), alto_three_musicmaker_one],
1147         [(46, 4), (47, 4), alto_three_musicmaker_two],
1148         [(47, 4), (48, 4), alto_three_musicmaker_three],
1149         [(49, 4), (50, 4), alto_three_musicmaker_one],
1150         [(50, 4), (51, 4), alto_three_musicmaker_one],
1151         [(51, 4), (52, 4), alto_three_musicmaker_two],
1152         [(52, 4), (53, 4), alto_three_musicmaker_two],
1153         [(53, 4), (54, 4), alto_three_musicmaker_two],
1154         [(54, 4), (55, 4), alto_three_musicmaker_two],

```

```

1155     [(56, 4), (57, 4), alto_three_musicmaker_one],
1156     [(57, 4), (58, 4), alto_three_musicmaker_one],
1157     [(58, 4), (59, 4), alto_three_musicmaker_one],
1158     [(59, 4), (60, 4), alto_three_musicmaker_one],
1159     [(60, 4), (61, 4), alto_three_musicmaker_one],
1160     [(61, 4), (62, 4), alto_three_musicmaker_one],
1161     [(67, 4), (68, 4), alto_three_musicmaker_two],
1162     [(68, 4), (69, 4), alto_three_musicmaker_two],
1163     [(69, 4), (70, 4), alto_three_musicmaker_two],
1164     [(70, 4), (71, 4), alto_three_musicmaker_two],
1165     [(74, 4), (75, 4), alto_three_musicmaker_three],
1166     [(75, 4), (76, 4), alto_three_musicmaker_three],
1167     [(77, 4), (78, 4), alto_three_musicmaker_one],
1168     [(78, 4), (79, 4), alto_three_musicmaker_one],
1169     [(79, 4), (80, 4), alto_three_musicmaker_two],
1170     [(80, 4), (81, 4), alto_three_musicmaker_two],
1171     [(81, 4), (82, 4), alto_three_musicmaker_two],
1172     [(82, 4), (83, 4), alto_three_musicmaker_two],
1173     ],
1174   ]
1175 ])
1176
1177 voice_8_timestspan_list = abjad.TimespanList([
1178     abjad.AnnotatedTimespan(
1179         start_offset=start_offset,
1180         stop_offset=stop_offset,
1181         annotation=MusicSpecifier(
1182             music_maker=music_maker,
1183             voice_name='Voice 8',
1184         ),
1185     )
1186     for start_offset, stop_offset, music_maker in [
1187         [(0, 4), (1, 4), alto_four_musicmaker_two],
1188         [(1, 4), (2, 4), alto_four_musicmaker_two],
1189         [(2, 4), (3, 4), alto_four_musicmaker_two],
1190         [(3, 4), (4, 4), alto_four_musicmaker_two],
1191         [(4, 4), (5, 4), alto_four_musicmaker_two],
1192         [(9, 4), (10, 4), alto_four_musicmaker_one],
1193         [(10, 4), (11, 4), alto_four_musicmaker_one],
1194         [(11, 4), (12, 4), alto_four_musicmaker_one],
1195         [(14, 4), (15, 4), alto_four_musicmaker_one],
1196         [(21, 4), (22, 4), alto_four_musicmaker_two],
1197         [(22, 4), (23, 4), alto_four_musicmaker_two],
1198         [(23, 4), (24, 4), alto_four_musicmaker_one],
1199         [(24, 4), (25, 4), alto_four_musicmaker_one],
1200         [(25, 4), (26, 4), alto_four_musicmaker_one],
1201
1202         [(28, 4), (29, 4), alto_four_musicmaker_one],
1203         [(29, 4), (30, 4), alto_four_musicmaker_one],
1204         [(30, 4), (31, 4), alto_four_musicmaker_one],
1205         [(31, 4), (32, 4), alto_four_musicmaker_one],
1206         [(32, 4), (33, 4), alto_four_musicmaker_one],
1207         [(37, 4), (38, 4), alto_four_musicmaker_two],
1208         [(38, 4), (39, 4), alto_four_musicmaker_two],
1209         [(39, 4), (40, 4), alto_four_musicmaker_two],
1210         [(42, 4), (43, 4), alto_four_musicmaker_three],
1211         [(49, 4), (50, 4), alto_four_musicmaker_one],
1212         [(50, 4), (51, 4), alto_four_musicmaker_one],
1213         [(51, 4), (52, 4), alto_four_musicmaker_two],

```

```

1214     [(52, 4), (53, 4), alto_four_musicmaker_two],
1215     [(53, 4), (54, 4), alto_four_musicmaker_two],
1216
1217     [(56, 4), (57, 4), alto_four_musicmaker_one],
1218     [(57, 4), (58, 4), alto_four_musicmaker_one],
1219     [(58, 4), (59, 4), alto_four_musicmaker_one],
1220     [(59, 4), (60, 4), alto_four_musicmaker_one],
1221     [(60, 4), (61, 4), alto_four_musicmaker_one],
1222     [(65, 4), (66, 4), alto_four_musicmaker_one],
1223     [(66, 4), (67, 4), alto_four_musicmaker_one],
1224     [(67, 4), (68, 4), alto_four_musicmaker_one],
1225     [(70, 4), (71, 4), alto_four_musicmaker_two],
1226     [(77, 4), (78, 4), alto_four_musicmaker_two],
1227     [(78, 4), (79, 4), alto_four_musicmaker_two],
1228     [(79, 4), (80, 4), alto_four_musicmaker_three],
1229     [(80, 4), (81, 4), alto_four_musicmaker_three],
1230     [(81, 4), (82, 4), alto_four_musicmaker_three],
1231 ]
1232 ])
1233
1234 voice_9_timespan_list = abjad.TimespanList([
1235     abjad.AnnotatedTimespan(
1236         start_offset=start_offset,
1237         stop_offset=stop_offset,
1238         annotation=MusicSpecifier(
1239             music_maker=music_maker,
1240             voice_name='Voice 9',
1241         ),
1242     )
1243     for start_offset, stop_offset, music_maker in [
1244         [(0, 4), (1, 4), alto_five_musicmaker_one],
1245         [(1, 4), (2, 4), alto_five_musicmaker_one],
1246         [(2, 4), (3, 4), alto_five_musicmaker_one],
1247         [(3, 4), (4, 4), alto_five_musicmaker_one],
1248         [(7, 4), (8, 4), alto_five_musicmaker_one],
1249         [(8, 4), (9, 4), alto_five_musicmaker_one],
1250         [(10, 4), (11, 4), alto_five_musicmaker_two],
1251         [(11, 4), (12, 4), alto_five_musicmaker_two],
1252         [(12, 4), (13, 4), alto_five_musicmaker_one],
1253         [(13, 4), (14, 4), alto_five_musicmaker_one],
1254         [(14, 4), (15, 4), alto_five_musicmaker_one],
1255         [(15, 4), (16, 4), alto_five_musicmaker_one],
1256         [(21, 4), (22, 4), alto_five_musicmaker_two],
1257         [(22, 4), (23, 4), alto_five_musicmaker_two],
1258         [(23, 4), (24, 4), alto_five_musicmaker_two],
1259         [(24, 4), (25, 4), alto_five_musicmaker_two],
1260
1261         [(28, 4), (29, 4), alto_five_musicmaker_one],
1262         [(29, 4), (30, 4), alto_five_musicmaker_one],
1263         [(30, 4), (31, 4), alto_five_musicmaker_one],
1264         [(31, 4), (32, 4), alto_five_musicmaker_one],
1265         [(35, 4), (36, 4), alto_five_musicmaker_two],
1266         [(36, 4), (37, 4), alto_five_musicmaker_two],
1267         [(38, 4), (39, 4), alto_five_musicmaker_three],
1268         [(39, 4), (40, 4), alto_five_musicmaker_three],
1269         [(40, 4), (41, 4), alto_five_musicmaker_one],
1270         [(41, 4), (42, 4), alto_five_musicmaker_one],
1271         [(42, 4), (43, 4), alto_five_musicmaker_one],
1272         [(43, 4), (44, 4), alto_five_musicmaker_one],

```

```

1273     [(49, 4), (50, 4), alto_five_musicmaker_two],
1274     [(50, 4), (51, 4), alto_five_musicmaker_two],
1275     [(51, 4), (52, 4), alto_five_musicmaker_three],
1276     [(52, 4), (53, 4), alto_five_musicmaker_three],
1277
1278     [(56, 4), (57, 4), alto_five_musicmaker_one],
1279     [(57, 4), (58, 4), alto_five_musicmaker_one],
1280     [(58, 4), (59, 4), alto_five_musicmaker_one],
1281     [(59, 4), (60, 4), alto_five_musicmaker_one],
1282     [(63, 4), (64, 4), alto_five_musicmaker_one],
1283     [(64, 4), (65, 4), alto_five_musicmaker_one],
1284     [(66, 4), (67, 4), alto_five_musicmaker_two],
1285     [(67, 4), (68, 4), alto_five_musicmaker_two],
1286     [(68, 4), (69, 4), alto_five_musicmaker_two],
1287     [(69, 4), (70, 4), alto_five_musicmaker_two],
1288     [(70, 4), (71, 4), alto_five_musicmaker_two],
1289     [(71, 4), (72, 4), alto_five_musicmaker_two],
1290     [(77, 4), (79, 4), alto_five_musicmaker_three],
1291     [(79, 4), (80, 4), alto_five_musicmaker_three],
1292     [(80, 4), (81, 4), alto_five_musicmaker_three],
1293 ]
1294 ])
1295
1296 voice_10_timespan_list = abjad.TimespanList([
1297     abjad.AnnotatedTimespan(
1298         start_offset=start_offset,
1299         stop_offset=stop_offset,
1300         annotation=MusicSpecifier(
1301             music_maker=music_maker,
1302             voice_name='Voice 10',
1303         ),
1304     ),
1305     for start_offset, stop_offset, music_maker in [
1306         [(0, 4), (1, 4), alto_six_musicmaker_one],
1307         [(1, 4), (2, 4), alto_six_musicmaker_one],
1308         [(2, 4), (3, 4), alto_six_musicmaker_one],
1309         [(5, 4), (6, 4), alto_six_musicmaker_two],
1310         [(12, 4), (13, 4), alto_six_musicmaker_one],
1311         [(13, 4), (14, 4), alto_six_musicmaker_one],
1312         [(14, 4), (15, 4), alto_six_musicmaker_one],
1313         [(15, 4), (16, 4), alto_six_musicmaker_one],
1314         [(16, 4), (17, 4), alto_six_musicmaker_two],
1315         [(21, 4), (22, 4), alto_six_musicmaker_two],
1316         [(22, 4), (23, 4), alto_six_musicmaker_two],
1317         [(23, 4), (24, 4), alto_six_musicmaker_one],
1318         [(26, 4), (27, 4), alto_six_musicmaker_two],
1319         [(27, 4), (28, 4), alto_six_musicmaker_two],
1320
1321         [(28, 4), (29, 4), alto_six_musicmaker_one],
1322         [(29, 4), (30, 4), alto_six_musicmaker_one],
1323         [(30, 4), (31, 4), alto_six_musicmaker_one],
1324         [(33, 4), (34, 4), alto_six_musicmaker_two],
1325         [(40, 4), (41, 4), alto_six_musicmaker_three],
1326         [(41, 4), (42, 4), alto_six_musicmaker_three],
1327         [(42, 4), (43, 4), alto_six_musicmaker_three],
1328         [(43, 4), (44, 4), alto_six_musicmaker_three],
1329         [(44, 4), (45, 4), alto_six_musicmaker_one],
1330         [(48, 4), (49, 4), alto_six_musicmaker_two],
1331         [(49, 4), (50, 4), alto_six_musicmaker_two],

```

```

1332     [(51, 4), (52, 4), alto_six_musicmaker_three],
1333     [(54, 4), (55, 4), alto_six_musicmaker_one],
1334
1335     [(56, 4), (57, 4), alto_six_musicmaker_one],
1336     [(57, 4), (58, 4), alto_six_musicmaker_one],
1337     [(58, 4), (59, 4), alto_six_musicmaker_one],
1338     [(61, 4), (62, 4), alto_six_musicmaker_one],
1339     [(68, 4), (69, 4), alto_six_musicmaker_two],
1340     [(69, 4), (70, 4), alto_six_musicmaker_two],
1341     [(70, 4), (71, 4), alto_six_musicmaker_two],
1342     [(71, 4), (72, 4), alto_six_musicmaker_two],
1343     [(72, 4), (73, 4), alto_six_musicmaker_two],
1344     [(76, 4), (77, 4), alto_six_musicmaker_three],
1345     [(77, 4), (78, 4), alto_six_musicmaker_three],
1346     [(78, 4), (79, 4), alto_six_musicmaker_three],
1347     [(82, 4), (83, 4), alto_six_musicmaker_one],
1348 ]
1349 ])
1350
1351 voice_11_timespan_list = abjad.TimespanList([
1352     abjad.AnnotatedTimespan(
1353         start_offset=start_offset,
1354         stop_offset=stop_offset,
1355         annotation=MusicSpecifier(
1356             music_maker=music_maker,
1357             voice_name='Voice 11',
1358         ),
1359     )
1360     for start_offset, stop_offset, music_maker in [
1361         [(0, 4), (1, 4), tenor_one_musicmaker_two],
1362         [(1, 4), (2, 4), tenor_one_musicmaker_two],
1363         [(3, 4), (4, 4), tenor_one_musicmaker_one],
1364         [(4, 4), (5, 4), tenor_one_musicmaker_one],
1365         [(5, 4), (6, 4), tenor_one_musicmaker_two],
1366         [(6, 4), (7, 4), tenor_one_musicmaker_two],
1367         [(7, 4), (8, 4), tenor_one_musicmaker_two],
1368         [(8, 4), (9, 4), tenor_one_musicmaker_two],
1369         [(14, 4), (15, 4), tenor_one_musicmaker_two],
1370         [(15, 4), (16, 4), tenor_one_musicmaker_two],
1371         [(16, 4), (17, 4), tenor_one_musicmaker_one],
1372         [(17, 4), (18, 4), tenor_one_musicmaker_one],
1373         [(21, 4), (22, 4), tenor_one_musicmaker_two],
1374         [(22, 4), (23, 4), tenor_one_musicmaker_two],
1375         [(24, 4), (25, 4), tenor_one_musicmaker_one],
1376         [(25, 4), (26, 4), tenor_one_musicmaker_one],
1377         [(26, 4), (27, 4), tenor_one_musicmaker_one],
1378
1379         [(28, 4), (29, 4), tenor_one_musicmaker_one],
1380         [(29, 4), (30, 4), tenor_one_musicmaker_one],
1381         [(31, 4), (32, 4), tenor_one_musicmaker_two],
1382         [(32, 4), (33, 4), tenor_one_musicmaker_two],
1383         [(33, 4), (34, 4), tenor_one_musicmaker_three],
1384         [(34, 4), (35, 4), tenor_one_musicmaker_three],
1385         [(35, 4), (36, 4), tenor_one_musicmaker_three],
1386         [(36, 4), (37, 4), tenor_one_musicmaker_three],
1387         [(42, 4), (43, 4), tenor_one_musicmaker_one],
1388         [(43, 4), (44, 4), tenor_one_musicmaker_one],
1389         [(44, 4), (45, 4), tenor_one_musicmaker_two],
1390         [(45, 4), (46, 4), tenor_one_musicmaker_two],

```

```

1391     [(49, 4), (50, 4), tenor_one_musicmaker_three],
1392     [(50, 4), (51, 4), tenor_one_musicmaker_three],
1393     [(52, 4), (53, 4), tenor_one_musicmaker_one],
1394     [(53, 4), (54, 4), tenor_one_musicmaker_one],
1395     [(54, 4), (55, 4), tenor_one_musicmaker_one],
1396
1397     [(56, 4), (57, 4), tenor_one_musicmaker_one],
1398     [(57, 4), (58, 4), tenor_one_musicmaker_one],
1399     [(59, 4), (60, 4), tenor_one_musicmaker_one],
1400     [(60, 4), (61, 4), tenor_one_musicmaker_one],
1401     [(61, 4), (62, 4), tenor_one_musicmaker_two],
1402     [(62, 4), (63, 4), tenor_one_musicmaker_two],
1403     [(63, 4), (64, 4), tenor_one_musicmaker_two],
1404     [(64, 4), (65, 4), tenor_one_musicmaker_two],
1405     [(70, 4), (71, 4), tenor_one_musicmaker_two],
1406     [(71, 4), (72, 4), tenor_one_musicmaker_two],
1407     [(72, 4), (73, 4), tenor_one_musicmaker_three],
1408     [(73, 4), (74, 4), tenor_one_musicmaker_three],
1409     [(77, 4), (78, 4), tenor_one_musicmaker_three],
1410     [(78, 4), (79, 4), tenor_one_musicmaker_three],
1411     [(80, 4), (81, 4), tenor_one_musicmaker_one],
1412     [(81, 4), (82, 4), tenor_one_musicmaker_one],
1413     [(82, 4), (83, 4), tenor_one_musicmaker_one],
1414 ]
1415 ])
1416
1417 voice_12_timespan_list = abjad.TimespanList([
1418     abjad.AnnotatedTimespan(
1419         start_offset=start_offset,
1420         stop_offset=stop_offset,
1421         annotation=MusicSpecifier(
1422             music_maker=music_maker,
1423             voice_name='Voice 12',
1424         ),
1425     )
1426     for start_offset, stop_offset, music_maker in [
1427         [(0, 4), (1, 4), tenor_two_musicmaker_one],
1428         [(7, 4), (8, 4), tenor_two_musicmaker_two],
1429         [(8, 4), (9, 4), tenor_two_musicmaker_two],
1430         [(9, 4), (10, 4), tenor_two_musicmaker_two],
1431         [(10, 4), (11, 4), tenor_two_musicmaker_two],
1432         [(11, 4), (12, 4), tenor_two_musicmaker_two],
1433         [(16, 4), (17, 4), tenor_two_musicmaker_one],
1434         [(17, 4), (18, 4), tenor_two_musicmaker_one],
1435         [(18, 4), (19, 4), tenor_two_musicmaker_one],
1436         [(21, 4), (22, 4), tenor_two_musicmaker_two],
1437
1438         [(28, 4), (29, 4), tenor_two_musicmaker_one],
1439         [(35, 4), (36, 4), tenor_two_musicmaker_two],
1440         [(36, 4), (37, 4), tenor_two_musicmaker_two],
1441         [(37, 4), (38, 4), tenor_two_musicmaker_three],
1442         [(38, 4), (39, 4), tenor_two_musicmaker_three],
1443         [(39, 4), (40, 4), tenor_two_musicmaker_three],
1444         [(44, 4), (45, 4), tenor_two_musicmaker_one],
1445         [(45, 4), (46, 4), tenor_two_musicmaker_one],
1446         [(46, 4), (47, 4), tenor_two_musicmaker_one],
1447         [(49, 4), (50, 4), tenor_two_musicmaker_two],
1448
1449         [(56, 4), (57, 4), tenor_two_musicmaker_one],

```

```

1450     [(63, 4), (64, 4), tenor_two_musicmaker_one],
1451     [(64, 4), (65, 4), tenor_two_musicmaker_one],
1452     [(65, 4), (66, 4), tenor_two_musicmaker_two],
1453     [(66, 4), (67, 4), tenor_two_musicmaker_two],
1454     [(67, 4), (68, 4), tenor_two_musicmaker_two],
1455     [(72, 4), (73, 4), tenor_two_musicmaker_two],
1456     [(73, 4), (74, 4), tenor_two_musicmaker_two],
1457     [(74, 4), (75, 4), tenor_two_musicmaker_two],
1458     [(77, 4), (78, 4), tenor_two_musicmaker_three],
1459   ]
1460 ])
1461
1462 voice_13_timespan_list = abjad.TimespanList([
1463     abjad.AnnotatedTimespan(
1464         start_offset=start_offset,
1465         stop_offset=stop_offset,
1466         annotation=MusicSpecifier(
1467             music_maker=music_maker,
1468             voice_name='Voice 13',
1469         ),
1470     )
1471     for start_offset, stop_offset, music_maker in [
1472         [(0, 4), (1, 4), tenor_three_musicmaker_two],
1473         [(1, 4), (2, 4), tenor_three_musicmaker_two],
1474         [(2, 4), (3, 4), tenor_three_musicmaker_two],
1475         [(3, 4), (4, 4), tenor_three_musicmaker_two],
1476         [(4, 4), (5, 4), tenor_three_musicmaker_two],
1477         [(5, 4), (6, 4), tenor_three_musicmaker_two],
1478         [(11, 4), (12, 4), tenor_three_musicmaker_one],
1479         [(12, 4), (13, 4), tenor_three_musicmaker_two],
1480         [(13, 4), (14, 4), tenor_three_musicmaker_two],
1481         [(14, 4), (15, 4), tenor_three_musicmaker_two],
1482         [(18, 4), (19, 4), tenor_three_musicmaker_one],
1483         [(19, 4), (20, 4), tenor_three_musicmaker_one],
1484         [(21, 4), (22, 4), tenor_three_musicmaker_two],
1485         [(22, 4), (23, 4), tenor_three_musicmaker_two],
1486         [(23, 4), (24, 4), tenor_three_musicmaker_one],
1487         [(24, 4), (25, 4), tenor_three_musicmaker_one],
1488         [(25, 4), (26, 4), tenor_three_musicmaker_one],
1489         [(26, 4), (27, 4), tenor_three_musicmaker_one],
1490
1491         [(28, 4), (29, 4), tenor_three_musicmaker_one],
1492         [(29, 4), (30, 4), tenor_three_musicmaker_one],
1493         [(30, 4), (31, 4), tenor_three_musicmaker_one],
1494         [(32, 4), (33, 4), tenor_three_musicmaker_one],
1495         [(33, 4), (34, 4), tenor_three_musicmaker_two],
1496         [(39, 4), (40, 4), tenor_three_musicmaker_three],
1497         [(40, 4), (41, 4), tenor_three_musicmaker_one],
1498         [(41, 4), (42, 4), tenor_three_musicmaker_one],
1499         [(42, 4), (43, 4), tenor_three_musicmaker_one],
1500         [(46, 4), (47, 4), tenor_three_musicmaker_two],
1501         [(47, 4), (48, 4), tenor_three_musicmaker_three],
1502         [(49, 4), (50, 4), tenor_three_musicmaker_one],
1503         [(50, 4), (51, 4), tenor_three_musicmaker_one],
1504         [(51, 4), (52, 4), tenor_three_musicmaker_two],
1505         [(52, 4), (53, 4), tenor_three_musicmaker_two],
1506         [(53, 4), (54, 4), tenor_three_musicmaker_two],
1507         [(54, 4), (55, 4), tenor_three_musicmaker_two],
1508     ]

```

```

1509     [(56, 4), (57, 4), tenor_three_musicmaker_one],
1510     [(57, 4), (58, 4), tenor_three_musicmaker_one],
1511     [(58, 4), (59, 4), tenor_three_musicmaker_one],
1512     [(59, 4), (60, 4), tenor_three_musicmaker_one],
1513     [(60, 4), (61, 4), tenor_three_musicmaker_one],
1514     [(61, 4), (62, 4), tenor_three_musicmaker_one],
1515     [(67, 4), (68, 4), tenor_three_musicmaker_two],
1516     [(68, 4), (69, 4), tenor_three_musicmaker_two],
1517     [(69, 4), (70, 4), tenor_three_musicmaker_two],
1518     [(70, 4), (71, 4), tenor_three_musicmaker_two],
1519     [(74, 4), (75, 4), tenor_three_musicmaker_three],
1520     [(75, 4), (76, 4), tenor_three_musicmaker_three],
1521     [(77, 4), (78, 4), tenor_three_musicmaker_one],
1522     [(78, 4), (79, 4), tenor_three_musicmaker_one],
1523     [(79, 4), (80, 4), tenor_three_musicmaker_two],
1524     [(80, 4), (81, 4), tenor_three_musicmaker_two],
1525     [(81, 4), (82, 4), tenor_three_musicmaker_two],
1526     [(82, 4), (83, 4), tenor_three_musicmaker_two],
1527 ]
1528 ])
1529
1530 voice_14_timespan_list = abjad.TimespanList([
1531     abjad.AnnotatedTimespan(
1532         start_offset=start_offset,
1533         stop_offset=stop_offset,
1534         annotation=MusicSpecifier(
1535             music_maker=music_maker,
1536             voice_name='Voice 14',
1537         ),
1538     )
1539     for start_offset, stop_offset, music_maker in [
1540         [(0, 4), (1, 4), tenor_four_musicmaker_two],
1541         [(1, 4), (2, 4), tenor_four_musicmaker_two],
1542         [(2, 4), (3, 4), tenor_four_musicmaker_two],
1543         [(3, 4), (4, 4), tenor_four_musicmaker_two],
1544         [(4, 4), (5, 4), tenor_four_musicmaker_two],
1545         [(9, 4), (10, 4), tenor_four_musicmaker_one],
1546         [(10, 4), (11, 4), tenor_four_musicmaker_one],
1547         [(11, 4), (12, 4), tenor_four_musicmaker_one],
1548         [(14, 4), (15, 4), tenor_four_musicmaker_two],
1549         [(21, 4), (22, 4), tenor_four_musicmaker_one],
1550         [(22, 4), (23, 4), tenor_four_musicmaker_one],
1551         [(23, 4), (24, 4), tenor_four_musicmaker_one],
1552         [(24, 4), (25, 4), tenor_four_musicmaker_one],
1553         [(25, 4), (26, 4), tenor_four_musicmaker_one],
1554
1555         [(28, 4), (29, 4), tenor_four_musicmaker_one],
1556         [(29, 4), (30, 4), tenor_four_musicmaker_one],
1557         [(30, 4), (31, 4), tenor_four_musicmaker_one],
1558         [(31, 4), (32, 4), tenor_four_musicmaker_one],
1559         [(32, 4), (33, 4), tenor_four_musicmaker_one],
1560         [(37, 4), (38, 4), tenor_four_musicmaker_two],
1561         [(38, 4), (39, 4), tenor_four_musicmaker_two],
1562         [(39, 4), (40, 4), tenor_four_musicmaker_two],
1563         [(42, 4), (43, 4), tenor_four_musicmaker_three],
1564         [(49, 4), (50, 4), tenor_four_musicmaker_one],
1565         [(50, 4), (51, 4), tenor_four_musicmaker_one],
1566         [(51, 4), (52, 4), tenor_four_musicmaker_two],
1567         [(52, 4), (53, 4), tenor_four_musicmaker_two],

```

```

1568     [(53, 4), (54, 4), tenor_four_musicmaker_two],
1569
1570     [(56, 4), (57, 4), tenor_four_musicmaker_one],
1571     [(57, 4), (58, 4), tenor_four_musicmaker_one],
1572     [(58, 4), (59, 4), tenor_four_musicmaker_one],
1573     [(59, 4), (60, 4), tenor_four_musicmaker_one],
1574     [(60, 4), (61, 4), tenor_four_musicmaker_one],
1575     [(65, 4), (66, 4), tenor_four_musicmaker_one],
1576     [(66, 4), (67, 4), tenor_four_musicmaker_one],
1577     [(67, 4), (68, 4), tenor_four_musicmaker_one],
1578     [(70, 4), (71, 4), tenor_four_musicmaker_two],
1579     [(77, 4), (78, 4), tenor_four_musicmaker_two],
1580     [(78, 4), (79, 4), tenor_four_musicmaker_two],
1581     [(79, 4), (80, 4), tenor_four_musicmaker_three],
1582     [(80, 4), (81, 4), tenor_four_musicmaker_three],
1583     [(81, 4), (82, 4), tenor_four_musicmaker_three],
1584 ]
1585 ])
1586
1587 voice_15_timespan_list = abjad.TimespanList([
1588     abjad.AnnotatedTimespan(
1589         start_offset=start_offset,
1590         stop_offset=stop_offset,
1591         annotation=MusicSpecifier(
1592             music_maker=music_maker,
1593             voice_name='Voice 15',
1594         ),
1595     )
1596     for start_offset, stop_offset, music_maker in [
1597         [(0, 4), (1, 4), tenor_five_musicmaker_one],
1598         [(1, 4), (2, 4), tenor_five_musicmaker_one],
1599         [(2, 4), (3, 4), tenor_five_musicmaker_one],
1600         [(3, 4), (4, 4), tenor_five_musicmaker_one],
1601         [(7, 4), (8, 4), tenor_five_musicmaker_two],
1602         [(8, 4), (9, 4), tenor_five_musicmaker_two],
1603         [(10, 4), (11, 4), tenor_five_musicmaker_one],
1604         [(11, 4), (12, 4), tenor_five_musicmaker_one],
1605         [(12, 4), (13, 4), tenor_five_musicmaker_one],
1606         [(13, 4), (14, 4), tenor_five_musicmaker_one],
1607         [(14, 4), (15, 4), tenor_five_musicmaker_one],
1608         [(15, 4), (16, 4), tenor_five_musicmaker_one],
1609         [(21, 4), (22, 4), tenor_five_musicmaker_two],
1610         [(22, 4), (23, 4), tenor_five_musicmaker_two],
1611         [(23, 4), (24, 4), tenor_five_musicmaker_one],
1612         [(24, 4), (25, 4), tenor_five_musicmaker_one],
1613
1614         [(28, 4), (29, 4), tenor_five_musicmaker_one],
1615         [(29, 4), (30, 4), tenor_five_musicmaker_one],
1616         [(30, 4), (31, 4), tenor_five_musicmaker_one],
1617         [(31, 4), (32, 4), tenor_five_musicmaker_one],
1618         [(35, 4), (36, 4), tenor_five_musicmaker_two],
1619         [(36, 4), (37, 4), tenor_five_musicmaker_two],
1620         [(38, 4), (39, 4), tenor_five_musicmaker_three],
1621         [(39, 4), (40, 4), tenor_five_musicmaker_three],
1622         [(40, 4), (41, 4), tenor_five_musicmaker_one],
1623         [(41, 4), (42, 4), tenor_five_musicmaker_one],
1624         [(42, 4), (43, 4), tenor_five_musicmaker_one],
1625         [(43, 4), (44, 4), tenor_five_musicmaker_one],
1626         [(49, 4), (50, 4), tenor_five_musicmaker_two],

```

```

1627     [(50, 4), (51, 4), tenor_five_musicmaker_two],
1628     [(51, 4), (52, 4), tenor_five_musicmaker_three],
1629     [(52, 4), (53, 4), tenor_five_musicmaker_three],
1630
1631     [(56, 4), (57, 4), tenor_five_musicmaker_one],
1632     [(57, 4), (58, 4), tenor_five_musicmaker_one],
1633     [(58, 4), (59, 4), tenor_five_musicmaker_one],
1634     [(59, 4), (60, 4), tenor_five_musicmaker_one],
1635     [(63, 4), (64, 4), tenor_five_musicmaker_one],
1636     [(64, 4), (65, 4), tenor_five_musicmaker_one],
1637     [(66, 4), (67, 4), tenor_five_musicmaker_two],
1638     [(67, 4), (68, 4), tenor_five_musicmaker_two],
1639     [(68, 4), (69, 4), tenor_five_musicmaker_two],
1640     [(69, 4), (70, 4), tenor_five_musicmaker_two],
1641     [(70, 4), (71, 4), tenor_five_musicmaker_two],
1642     [(71, 4), (72, 4), tenor_five_musicmaker_two],
1643     [(77, 4), (78, 4), tenor_five_musicmaker_three],
1644     [(78, 4), (79, 4), tenor_five_musicmaker_three],
1645     [(79, 4), (80, 4), tenor_five_musicmaker_three],
1646     [(80, 4), (81, 4), tenor_five_musicmaker_three],
1647 ]
1648 ])
1649
1650 voice_16_timespan_list = abjad.TimespanList([
1651     abjad.AnnotatedTimespan(
1652         start_offset=start_offset,
1653         stop_offset=stop_offset,
1654         annotation=MusicSpecifier(
1655             music_maker=music_maker,
1656             voice_name='Voice 16',
1657         ),
1658     ),
1659     for start_offset, stop_offset, music_maker in [
1660         [(0, 4), (1, 4), baritone_one_musicmaker_two],
1661         [(1, 4), (2, 4), baritone_one_musicmaker_two],
1662         [(2, 4), (3, 4), baritone_one_musicmaker_two],
1663         [(5, 4), (6, 4), baritone_one_musicmaker_one],
1664         [(12, 4), (13, 4), baritone_one_musicmaker_one],
1665         [(13, 4), (14, 4), baritone_one_musicmaker_one],
1666         [(14, 4), (15, 4), baritone_one_musicmaker_one],
1667         [(15, 4), (16, 4), baritone_one_musicmaker_one],
1668         [(16, 4), (17, 4), baritone_one_musicmaker_two],
1669         [(21, 4), (22, 4), baritone_one_musicmaker_one],
1670         [(22, 4), (23, 4), baritone_one_musicmaker_one],
1671         [(23, 4), (24, 4), baritone_one_musicmaker_two],
1672         [(26, 4), (27, 4), baritone_one_musicmaker_two],
1673
1674         [(28, 4), (29, 4), baritone_one_musicmaker_one],
1675         [(29, 4), (30, 4), baritone_one_musicmaker_one],
1676         [(30, 4), (31, 4), baritone_one_musicmaker_one],
1677         [(33, 4), (34, 4), baritone_one_musicmaker_two],
1678         [(40, 4), (41, 4), baritone_one_musicmaker_three],
1679         [(41, 4), (42, 4), baritone_one_musicmaker_three],
1680         [(42, 4), (43, 4), baritone_one_musicmaker_three],
1681         [(43, 4), (44, 4), baritone_one_musicmaker_three],
1682         [(44, 4), (45, 4), baritone_one_musicmaker_one],
1683         [(49, 4), (50, 4), baritone_one_musicmaker_two],
1684         [(50, 4), (51, 4), baritone_one_musicmaker_two],
1685         [(51, 4), (52, 4), baritone_one_musicmaker_three],

```

```

1686     [(54, 4), (55, 4), baritone_one_musicmaker_one],
1687
1688     [(56, 4), (57, 4), baritone_one_musicmaker_one],
1689     [(57, 4), (58, 4), baritone_one_musicmaker_one],
1690     [(58, 4), (59, 4), baritone_one_musicmaker_one],
1691     [(61, 4), (62, 4), baritone_one_musicmaker_one],
1692     [(68, 4), (69, 4), baritone_one_musicmaker_two],
1693     [(69, 4), (70, 4), baritone_one_musicmaker_two],
1694     [(70, 4), (71, 4), baritone_one_musicmaker_two],
1695     [(71, 4), (72, 4), baritone_one_musicmaker_two],
1696     [(72, 4), (73, 4), baritone_one_musicmaker_two],
1697     [(77, 4), (78, 4), baritone_one_musicmaker_three],
1698     [(78, 4), (79, 4), baritone_one_musicmaker_three],
1699     [(79, 4), (80, 4), baritone_one_musicmaker_three],
1700     [(82, 4), (83, 4), baritone_one_musicmaker_one],
1701 ]
1702 ])
1703
1704 voice_17_timespan_list = abjad.TimespanList([
1705     abjad.AnnotatedTimespan(
1706         start_offset=start_offset,
1707         stop_offset=stop_offset,
1708         annotation=MusicSpecifier(
1709             music_maker=music_maker,
1710             voice_name='Voice 17',
1711         ),
1712     )
1713     for start_offset, stop_offset, music_maker in [
1714         [(0, 4), (1, 4), baritone_two_musicmaker_one],
1715         [(1, 4), (2, 4), baritone_two_musicmaker_one],
1716         [(3, 4), (4, 4), baritone_two_musicmaker_one],
1717         [(4, 4), (5, 4), baritone_two_musicmaker_one],
1718         [(5, 4), (6, 4), baritone_two_musicmaker_two],
1719         [(6, 4), (7, 4), baritone_two_musicmaker_two],
1720         [(7, 4), (8, 4), baritone_two_musicmaker_two],
1721         [(8, 4), (9, 4), baritone_two_musicmaker_two],
1722         [(14, 4), (15, 4), baritone_two_musicmaker_one],
1723         [(15, 4), (16, 4), baritone_two_musicmaker_one],
1724         [(16, 4), (17, 4), baritone_two_musicmaker_two],
1725         [(17, 4), (18, 4), baritone_two_musicmaker_two],
1726         [(21, 4), (22, 4), baritone_two_musicmaker_two],
1727         [(22, 4), (23, 4), baritone_two_musicmaker_two],
1728         [(24, 4), (25, 4), baritone_two_musicmaker_one],
1729         [(25, 4), (26, 4), baritone_two_musicmaker_one],
1730         [(26, 4), (27, 4), baritone_two_musicmaker_one],
1731
1732         [(28, 4), (29, 4), baritone_two_musicmaker_one],
1733         [(29, 4), (30, 4), baritone_two_musicmaker_one],
1734         [(31, 4), (32, 4), baritone_two_musicmaker_two],
1735         [(32, 4), (33, 4), baritone_two_musicmaker_two],
1736         [(33, 4), (34, 4), baritone_two_musicmaker_three],
1737         [(34, 4), (35, 4), baritone_two_musicmaker_three],
1738         [(35, 4), (36, 4), baritone_two_musicmaker_three],
1739         [(36, 4), (37, 4), baritone_two_musicmaker_three],
1740         [(42, 4), (43, 4), baritone_two_musicmaker_one],
1741         [(43, 4), (44, 4), baritone_two_musicmaker_one],
1742         [(44, 4), (45, 4), baritone_two_musicmaker_two],
1743         [(45, 4), (46, 4), baritone_two_musicmaker_two],
1744         [(49, 4), (50, 4), baritone_two_musicmaker_three],

```

```

1745     [(50, 4), (51, 4), baritone_two_musicmaker_three],
1746     [(52, 4), (53, 4), baritone_two_musicmaker_one],
1747     [(53, 4), (54, 4), baritone_two_musicmaker_one],
1748     [(54, 4), (55, 4), baritone_two_musicmaker_one],
1749
1750     [(56, 4), (57, 4), baritone_two_musicmaker_one],
1751     [(57, 4), (58, 4), baritone_two_musicmaker_one],
1752     [(59, 4), (60, 4), baritone_two_musicmaker_one],
1753     [(60, 4), (61, 4), baritone_two_musicmaker_one],
1754     [(61, 4), (62, 4), baritone_two_musicmaker_two],
1755     [(62, 4), (63, 4), baritone_two_musicmaker_two],
1756     [(63, 4), (64, 4), baritone_two_musicmaker_two],
1757     [(64, 4), (65, 4), baritone_two_musicmaker_two],
1758     [(70, 4), (71, 4), baritone_two_musicmaker_two],
1759     [(71, 4), (72, 4), baritone_two_musicmaker_two],
1760     [(72, 4), (73, 4), baritone_two_musicmaker_three],
1761     [(73, 4), (74, 4), baritone_two_musicmaker_three],
1762     [(77, 4), (78, 4), baritone_two_musicmaker_three],
1763     [(78, 4), (79, 4), baritone_two_musicmaker_three],
1764     [(80, 4), (81, 4), baritone_two_musicmaker_one],
1765     [(81, 4), (82, 4), baritone_two_musicmaker_one],
1766     [(82, 4), (83, 4), baritone_two_musicmaker_one],
1767 ]
1768 ])
1769
1770 voice_18_timespan_list = abjad.TimespanList([
1771     abjad.AnnotatedTimespan(
1772         start_offset=start_offset,
1773         stop_offset=stop_offset,
1774         annotation=MusicSpecifier(
1775             music_maker=music_maker,
1776             voice_name='Voice 18',
1777         ),
1778     ),
1779     for start_offset, stop_offset, music_maker in [
1780         [(0, 4), (1, 4), baritone_three_musicmaker_one],
1781         [(7, 4), (8, 4), baritone_three_musicmaker_two],
1782         [(8, 4), (9, 4), baritone_three_musicmaker_two],
1783         [(9, 4), (10, 4), baritone_three_musicmaker_one],
1784         [(10, 4), (11, 4), baritone_three_musicmaker_one],
1785         [(11, 4), (12, 4), baritone_three_musicmaker_one],
1786         [(16, 4), (17, 4), baritone_three_musicmaker_two],
1787         [(17, 4), (18, 4), baritone_three_musicmaker_two],
1788         [(18, 4), (19, 4), baritone_three_musicmaker_two],
1789         [(21, 4), (22, 4), baritone_three_musicmaker_two],
1790
1791         [(28, 4), (29, 4), baritone_three_musicmaker_one],
1792         [(35, 4), (36, 4), baritone_three_musicmaker_two],
1793         [(36, 4), (37, 4), baritone_three_musicmaker_two],
1794         [(37, 4), (38, 4), baritone_three_musicmaker_three],
1795         [(38, 4), (39, 4), baritone_three_musicmaker_three],
1796         [(39, 4), (40, 4), baritone_three_musicmaker_three],
1797         [(44, 4), (45, 4), baritone_three_musicmaker_one],
1798         [(45, 4), (46, 4), baritone_three_musicmaker_one],
1799         [(46, 4), (47, 4), baritone_three_musicmaker_one],
1800         [(49, 4), (50, 4), baritone_three_musicmaker_two],
1801
1802         [(56, 4), (57, 4), baritone_three_musicmaker_one],
1803         [(63, 4), (64, 4), baritone_three_musicmaker_one],

```

```

1804     [(64, 4), (65, 4), baritone_three_musicmaker_one],
1805     [(65, 4), (66, 4), baritone_three_musicmaker_two],
1806     [(66, 4), (67, 4), baritone_three_musicmaker_two],
1807     [(67, 4), (68, 4), baritone_three_musicmaker_two],
1808     [(72, 4), (73, 4), baritone_three_musicmaker_two],
1809     [(73, 4), (74, 4), baritone_three_musicmaker_two],
1810     [(74, 4), (75, 4), baritone_three_musicmaker_two],
1811     [(77, 4), (78, 4), baritone_three_musicmaker_three],
1812 ]
1813 ])
1814
1815 voice_19_timespan_list = abjad.TimespanList([
1816     abjad.AnnotatedTimespan(
1817         start_offset=start_offset,
1818         stop_offset=stop_offset,
1819         annotation=MusicSpecifier(
1820             music_maker=music_maker,
1821             voice_name='Voice 19',
1822         ),
1823     )
1824     for start_offset, stop_offset, music_maker in [
1825         [(0, 4), (1, 4), bass_one_musicmaker_two],
1826         [(1, 4), (2, 4), bass_one_musicmaker_two],
1827         [(2, 4), (3, 4), bass_one_musicmaker_two],
1828         [(3, 4), (4, 4), bass_one_musicmaker_two],
1829         [(4, 4), (5, 4), bass_one_musicmaker_two],
1830         [(5, 4), (6, 4), bass_one_musicmaker_one],
1831         [(11, 4), (12, 4), bass_one_musicmaker_two],
1832         [(12, 4), (13, 4), bass_one_musicmaker_two],
1833         [(13, 4), (14, 4), bass_one_musicmaker_two],
1834         [(14, 4), (15, 4), bass_one_musicmaker_two],
1835         [(18, 4), (19, 4), bass_one_musicmaker_one],
1836         [(19, 4), (20, 4), bass_one_musicmaker_two],
1837         [(21, 4), (22, 4), bass_one_musicmaker_one],
1838         [(22, 4), (23, 4), bass_one_musicmaker_one],
1839         [(23, 4), (24, 4), bass_one_musicmaker_one],
1840         [(24, 4), (25, 4), bass_one_musicmaker_one],
1841         [(25, 4), (26, 4), bass_one_musicmaker_one],
1842         [(26, 4), (27, 4), bass_one_musicmaker_one],
1843         [(28, 4), (29, 4), bass_one_musicmaker_one],
1844         [(29, 4), (30, 4), bass_one_musicmaker_one],
1845         [(30, 4), (31, 4), bass_one_musicmaker_one],
1846         [(31, 4), (32, 4), bass_one_musicmaker_one],
1847         [(32, 4), (33, 4), bass_one_musicmaker_one],
1848         [(33, 4), (34, 4), bass_one_musicmaker_two],
1849         [(39, 4), (40, 4), bass_one_musicmaker_three],
1850         [(40, 4), (41, 4), bass_one_musicmaker_one],
1851         [(41, 4), (42, 4), bass_one_musicmaker_one],
1852         [(42, 4), (43, 4), bass_one_musicmaker_one],
1853         [(46, 4), (47, 4), bass_one_musicmaker_two],
1854         [(47, 4), (48, 4), bass_one_musicmaker_three],
1855         [(49, 4), (50, 4), bass_one_musicmaker_one],
1856         [(50, 4), (51, 4), bass_one_musicmaker_one],
1857         [(51, 4), (52, 4), bass_one_musicmaker_two],
1858         [(52, 4), (53, 4), bass_one_musicmaker_two],
1859         [(53, 4), (54, 4), bass_one_musicmaker_two],
1860         [(54, 4), (55, 4), bass_one_musicmaker_two],
1861
1862

```

```

1863     [(56, 4), (57, 4), bass_one_musicmaker_one],
1864     [(57, 4), (58, 4), bass_one_musicmaker_one],
1865     [(58, 4), (59, 4), bass_one_musicmaker_one],
1866     [(59, 4), (60, 4), bass_one_musicmaker_one],
1867     [(60, 4), (61, 4), bass_one_musicmaker_one],
1868     [(61, 4), (62, 4), bass_one_musicmaker_one],
1869     [(67, 4), (68, 4), bass_one_musicmaker_two],
1870     [(68, 4), (69, 4), bass_one_musicmaker_two],
1871     [(69, 4), (70, 4), bass_one_musicmaker_two],
1872     [(70, 4), (71, 4), bass_one_musicmaker_two],
1873     [(74, 4), (75, 4), bass_one_musicmaker_three],
1874     [(75, 4), (76, 4), bass_one_musicmaker_three],
1875     [(77, 4), (78, 4), bass_one_musicmaker_one],
1876     [(78, 4), (79, 4), bass_one_musicmaker_one],
1877     [(79, 4), (80, 4), bass_one_musicmaker_two],
1878     [(80, 4), (81, 4), bass_one_musicmaker_two],
1879     [(81, 4), (82, 4), bass_one_musicmaker_two],
1880     [(82, 4), (83, 4), bass_one_musicmaker_two],
1881 ]
1882 ])
1883
1884 voice_20_timespan_list = abjad.TimespanList([
1885     abjad.AnnotatedTimespan(
1886         start_offset=start_offset,
1887         stop_offset=stop_offset,
1888         annotation=MusicSpecifier(
1889             music_maker=music_maker,
1890             voice_name='Voice 20',
1891         ),
1892     )
1893     for start_offset, stop_offset, music_maker in [
1894         [(0, 4), (1, 4), bass_two_musicmaker_one],
1895         [(1, 4), (2, 4), bass_two_musicmaker_one],
1896         [(2, 4), (3, 4), bass_two_musicmaker_one],
1897         [(3, 4), (4, 4), bass_two_musicmaker_one],
1898         [(4, 4), (5, 4), bass_two_musicmaker_one],
1899         [(9, 4), (10, 4), bass_two_musicmaker_two],
1900         [(10, 4), (11, 4), bass_two_musicmaker_two],
1901         [(11, 4), (12, 4), bass_two_musicmaker_two],
1902         [(14, 4), (15, 4), bass_two_musicmaker_one],
1903         [(21, 4), (22, 4), bass_two_musicmaker_two],
1904         [(22, 4), (23, 4), bass_two_musicmaker_two],
1905         [(23, 4), (24, 4), bass_two_musicmaker_two],
1906         [(24, 4), (25, 4), bass_two_musicmaker_two],
1907         [(25, 4), (26, 4), bass_two_musicmaker_two],
1908
1909         [(28, 4), (29, 4), bass_two_musicmaker_one],
1910         [(29, 4), (30, 4), bass_two_musicmaker_one],
1911         [(30, 4), (31, 4), bass_two_musicmaker_one],
1912         [(31, 4), (32, 4), bass_two_musicmaker_one],
1913         [(32, 4), (33, 4), bass_two_musicmaker_one],
1914         [(37, 4), (38, 4), bass_two_musicmaker_two],
1915         [(38, 4), (39, 4), bass_two_musicmaker_two],
1916         [(39, 4), (40, 4), bass_two_musicmaker_two],
1917         [(42, 4), (43, 4), bass_two_musicmaker_three],
1918         [(49, 4), (50, 4), bass_two_musicmaker_one],
1919         [(50, 4), (51, 4), bass_two_musicmaker_one],
1920         [(51, 4), (52, 4), bass_two_musicmaker_two],
1921         [(52, 4), (53, 4), bass_two_musicmaker_two],

```

```

1922     [(53, 4), (54, 4), bass_two_musicmaker_two],
1923
1924     [(56, 4), (57, 4), bass_two_musicmaker_one],
1925     [(57, 4), (58, 4), bass_two_musicmaker_one],
1926     [(58, 4), (59, 4), bass_two_musicmaker_one],
1927     [(59, 4), (60, 4), bass_two_musicmaker_one],
1928     [(60, 4), (61, 4), bass_two_musicmaker_one],
1929     [(65, 4), (66, 4), bass_two_musicmaker_one],
1930     [(66, 4), (67, 4), bass_two_musicmaker_one],
1931     [(67, 4), (68, 4), bass_two_musicmaker_one],
1932     [(70, 4), (71, 4), bass_two_musicmaker_two],
1933     [(77, 4), (78, 4), bass_two_musicmaker_two],
1934     [(78, 4), (79, 4), bass_two_musicmaker_two],
1935     [(79, 4), (80, 4), bass_two_musicmaker_three],
1936     [(80, 4), (81, 4), bass_two_musicmaker_three],
1937     [(81, 4), (82, 4), bass_two_musicmaker_three],
1938 ]
1939 ])
1940
1941 voice_21_timespan_list = abjad.TimespanList([
1942     abjad.AnnotatedTimespan(
1943         start_offset=start_offset,
1944         stop_offset=stop_offset,
1945         annotation=MusicSpecifier(
1946             music_maker=music_maker,
1947             voice_name='Voice 21',
1948         ),
1949     )
1950     for start_offset, stop_offset, music_maker in [
1951         [(0, 4), (1, 4), contrabass_musicmaker_two],
1952         [(1, 4), (2, 4), contrabass_musicmaker_two],
1953         [(2, 4), (3, 4), contrabass_musicmaker_two],
1954         [(3, 4), (4, 4), contrabass_musicmaker_two],
1955         [(6, 4), (7, 4), contrabass_musicmaker_one],
1956         [(7, 4), (8, 4), contrabass_musicmaker_one],
1957         [(10, 4), (11, 4), contrabass_musicmaker_two],
1958         [(11, 4), (12, 4), contrabass_musicmaker_two],
1959         [(12, 4), (13, 4), contrabass_musicmaker_two],
1960         [(13, 4), (14, 4), contrabass_musicmaker_two],
1961         [(14, 4), (15, 4), contrabass_musicmaker_two],
1962         [(15, 4), (16, 4), contrabass_musicmaker_two],
1963         [(20, 4), (21, 4), contrabass_musicmaker_one],
1964         [(21, 4), (22, 4), contrabass_musicmaker_one],
1965         [(23, 4), (24, 4), contrabass_musicmaker_two],
1966         [(24, 4), (25, 4), contrabass_musicmaker_two],
1967
1968         [(28, 4), (29, 4), contrabass_musicmaker_one],
1969         [(29, 4), (30, 4), contrabass_musicmaker_one],
1970         [(30, 4), (31, 4), contrabass_musicmaker_one],
1971         [(31, 4), (32, 4), contrabass_musicmaker_one],
1972         [(35, 4), (36, 4), contrabass_musicmaker_two],
1973         [(36, 4), (37, 4), contrabass_musicmaker_two],
1974         [(38, 4), (39, 4), contrabass_musicmaker_three],
1975         [(39, 4), (40, 4), contrabass_musicmaker_three],
1976         [(40, 4), (41, 4), contrabass_musicmaker_one],
1977         [(41, 4), (42, 4), contrabass_musicmaker_one],
1978         [(42, 4), (43, 4), contrabass_musicmaker_one],
1979         [(43, 4), (44, 4), contrabass_musicmaker_one],
1980         [(48, 4), (49, 4), contrabass_musicmaker_two],

```

```

1981     [(49, 4), (50, 4), contrabass_musicmaker_two],
1982     [(51, 4), (52, 4), contrabass_musicmaker_three],
1983     [(52, 4), (53, 4), contrabass_musicmaker_three],
1984
1985     [(56, 4), (57, 4), contrabass_musicmaker_one],
1986     [(57, 4), (58, 4), contrabass_musicmaker_one],
1987     [(58, 4), (59, 4), contrabass_musicmaker_one],
1988     [(59, 4), (60, 4), contrabass_musicmaker_one],
1989     [(62, 4), (63, 4), contrabass_musicmaker_one],
1990     [(63, 4), (64, 4), contrabass_musicmaker_one],
1991     [(66, 4), (67, 4), contrabass_musicmaker_two],
1992     [(67, 4), (68, 4), contrabass_musicmaker_two],
1993     [(68, 4), (69, 4), contrabass_musicmaker_two],
1994     [(69, 4), (70, 4), contrabass_musicmaker_two],
1995     [(70, 4), (71, 4), contrabass_musicmaker_two],
1996     [(71, 4), (72, 4), contrabass_musicmaker_two],
1997     [(76, 4), (77, 4), contrabass_musicmaker_three],
1998     [(77, 4), (78, 4), contrabass_musicmaker_three],
1999     [(78, 4), (79, 4), contrabass_musicmaker_three],
2000     [(79, 4), (80, 4), contrabass_musicmaker_three],
2001 ]
2002 ])
2003
2004 all_timespan_lists = {
2005     'Voice 1': voice_1_timespan_list,
2006     'Voice 2': voice_2_timespan_list,
2007     'Voice 3': voice_3_timespan_list,
2008     'Voice 4': voice_4_timespan_list,
2009     'Voice 5': voice_5_timespan_list,
2010     'Voice 6': voice_6_timespan_list,
2011     'Voice 7': voice_7_timespan_list,
2012     'Voice 8': voice_8_timespan_list,
2013     'Voice 9': voice_9_timespan_list,
2014     'Voice 10': voice_10_timespan_list,
2015     'Voice 11': voice_11_timespan_list,
2016     'Voice 12': voice_12_timespan_list,
2017     'Voice 13': voice_13_timespan_list,
2018     'Voice 14': voice_14_timespan_list,
2019     'Voice 15': voice_15_timespan_list,
2020     'Voice 16': voice_16_timespan_list,
2021     'Voice 17': voice_17_timespan_list,
2022     'Voice 18': voice_18_timespan_list,
2023     'Voice 19': voice_19_timespan_list,
2024     'Voice 20': voice_20_timespan_list,
2025     'Voice 21': voice_21_timespan_list,
2026 }
2027
2028 global_timespan = abjad.Timespan(
2029     start_offset=0,
2030     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
2031 )
2032
2033 for voice_name, timespan_list in all_timespan_lists.items():
2034     silences = abjad.TimespanList([global_timespan])
2035     silences.extend(timespan_list)
2036     silences.sort()
2037     silences.compute_logical_xor()
2038     for silence_timespan in silences:
2039         timespan_list.append(

```

```

2040     abjad.AnnotatedTimespan(
2041         start_offset=silence_timespan.start_offset,
2042         stop_offset=silence_timespan.stop_offset,
2043         annotation=MusicSpecifier(
2044             music_maker=None,
2045             voice_name=voice_name,
2046         ),
2047     ),
2048 )
2049 timespan_list.sort()
2050
2051 for voice_name, timespan_list in all_timespan_lists.items():
2052     shards = timespan_list.split_at_offsets(bounds)
2053     split_timespan_list = abjad.TimespanList()
2054     for shard in shards:
2055         split_timespan_list.extend(shard)
2056     split_timespan_list.sort()
2057     all_timespan_lists[voice_name] = timespan_list
2058
2059 score = abjad.Score([
2060     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
2061     abjad.StaffGroup(
2062         [
2063             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
2064             lilypond_type='Staff',),
2065             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
2066             lilypond_type='Staff',),
2067             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
2068             lilypond_type='Staff',),
2069             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
2070             lilypond_type='Staff',),
2071             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
2072             lilypond_type='Staff',),
2073             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
2074             lilypond_type='Staff',),
2075             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
2076             lilypond_type='Staff',),
2077             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
2078             lilypond_type='Staff',),
2079             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
2080             lilypond_type='Staff',),
2081             abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
2082             lilypond_type='Staff',),
2083             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
2084             lilypond_type='Staff',),
2085             abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
2086             lilypond_type='Staff',),
2087             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
2088             lilypond_type='Staff',),
2089             abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
2090             lilypond_type='Staff',),
2091             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
2092             lilypond_type='Staff',),
2093             abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
2094             lilypond_type='Staff',),
2095             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
2096             lilypond_type='Staff',),
2097             abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
2098             lilypond_type='Staff',),

```

```

2081     abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
2082     lilypond_type='Staff',),
2083     abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
2084     lilypond_type='Staff',),
2085     abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
2086     lilypond_type='Staff',),
2087     ],
2088     name='Staff Group',
2089   )
2090 ])
2091
2092 for time_signature in time_signatures:
2093   skip = abjad.Skip(1, multiplier=(time_signature))
2094   abjad.attach(time_signature, skip)
2095   score['Global Context'].append(skip)
2096
2097 print('Making containers ...')
2098
2099 def make_container(music_maker, durations):
2100   selections = music_maker(durations)
2101   container = abjad.Container([])
2102   container.extend(selections)
2103   return container
2104
2105 def key_function(timespan):
2106   return timespan.annotation.music_maker or silence_maker
2107
2108 for voice_name, timespan_list in all_timespan_lists.items():
2109   for music_maker, grouper in itertools.groupby(
2110     timespan_list,
2111     key=key_function,
2112   ):
2113     durations = [timespan.duration for timespan in grouper]
2114     container = make_container(music_maker, durations)
2115     voice = score[voice_name]
2116     voice.append(container)
2117
2118 print('Splitting and rewriting ...')
2119 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
2120   for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
2121     time_signature = time_signatures[i]
2122     abjad.mutate(shard).rewrite_meter(time_signature)
2123
2124 print('Beaming runs ...')
2125 for voice in abjad.select(score).components(abjad.Voice):
2126   for run in abjad.select(voice).runs():
2127     specifier = abjadext.rmakers.BeamSpecifier(
2128       beam_each_division=False,
2129     )
2130     specifier(run)
2131     abjad.beam(voice[:], beam_lone_notes=False, beam_rests=False,)
2132
2133 print('Stopping Hairpins ...')
2134 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2135   for rest in abjad.iterate(staff).components(abjad.Rest):
2136     previous_leaf = abjad.inspect(rest).leaf(-1)
2137     if isinstance(previous_leaf, abjad.Note):
2138       abjad.attach(abjad.StopHairpin(), rest)
2139     elif isinstance(previous_leaf, abjad.Chord):

```

```

2137         abjad.attach(abjad.StopHairpin(), rest)
2138     elif isinstance(previous_leaf, abjad.Rest):
2139         pass
2140
2141 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2142     first_leaf = abjad.select(staff).leaves()[0]
2143     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
2144     abjad.attach(stop, first_leaf)
2145
2146 print('Adding attachments ...')
2147 bar_line = abjad.BarLine('||')
2148 # metro = abjad.MetronomeMark((1, 4), 90)
2149 markup = abjad.Markup(r'\bold { A }')
2150 mark = abjad.RehearsalMark(markup=markup)
2151
2152 size = abjad.MarkupCommand(
2153     'override',
2154     abjad.SchemePair(('size', .5)),
2155 )
2156
2157 fingering_1 = abjad.WoodwindFingering( #[13.5, 16, 26.5]
2158     name='soprano-saxophone',
2159     center_column=( 'one', 'two', 'three', 'four', 'six', ),
2160     left_hand=( 'ees', 'd', ),
2161     right_hand=( 'low-c', ),
2162 )
2163 command_1 = fingering_1()
2164 soprano_1_fingering = abjad.Markup(contents=[size, command_1], direction=abjad.Down)
2165 soprano_1_multiphonic = [x for x in abjad.iterate(score['Voice 2']).components(
2166     abjad.Chord)]
2167 abjad.attach(soprano_1_fingering, soprano_1_multiphonic[0])
2168
2169 fingering_2 = abjad.WoodwindFingering( #[13, 16, 26]
2170     name='soprano-saxophone',
2171     center_column=( 'one', 'two', 'three', 'four', 'six', ),
2172     left_hand=( 'ees', 'd', 'b', ),
2173     right_hand=( 'low-c', ),
2174 )
2175 command_2 = fingering_2()
2176 soprano_2_fingering = abjad.Markup(contents=[size, command_2], direction=abjad.Down)
2177 soprano_2_multiphonic = [x for x in abjad.iterate(score['Voice 3']).components(
2178     abjad.Chord)]
2179 abjad.attach(soprano_2_fingering, soprano_2_multiphonic[0])
2180
2181 fingering_3 = abjad.WoodwindFingering( #[12.5, 14.5, 26]
2182     name='soprano-saxophone',
2183     center_column=( 'one', 'two', 'three', 'four', 'five', ),
2184     left_hand=( 'ees', 'd', ),
2185     right_hand=( 'low-c', ),
2186 )
2187 command_3 = fingering_3()
2188 soprano_3_fingering = abjad.Markup(contents=[size, command_3], direction=abjad.Down)
2189 soprano_3_multiphonic = [x for x in abjad.iterate(score['Voice 4']).components(
2190     abjad.Chord)]
2191 abjad.attach(soprano_3_fingering, soprano_3_multiphonic[0])
2192

```

```

2190 fingering_4 = abjad.WoodwindFingering( #[12.5, 19, 27.5, 34]
2191     name='alto-saxophone',
2192     center_column=('one', 'two', 'three', ),
2193     left_hand=(d, gis, ),
2194     right_hand=(),
2195 )
2196 command_4 = fingering_4()
2197 alto_1_fingering = abjad.Markup(contents=[size, command_4], direction=abjad.Down)
2198 alto_1_multiphonic = [x for x in abjad.iterate(score['Voice 5']).components(abjad.
Chord)]
2199 abjad.attach(alto_1_fingering, alto_1_multiphonic[0])
2200 alto_6_multiphonic = [x for x in abjad.iterate(score['Voice 10']).components(abjad.
.Chord)]
2201 abjad.attach(alto_1_fingering, alto_6_multiphonic[0])
2202
2203 fingering_5 = abjad.WoodwindFingering( #[12.5, 15.5, 25.5]
2204     name='alto-saxophone',
2205     center_column=('one', 'two', 'three', 'four', 'five', ),
2206     left_hand=(b, ),
2207     right_hand=(low-c, c, ),
2208 )
2209 command_5 = fingering_5()
2210 alto_2_fingering = abjad.Markup(contents=[size, command_5], direction=abjad.Down)
2211 alto_2_multiphonic = [x for x in abjad.iterate(score['Voice 6']).components(abjad.
Chord)]
2212 abjad.attach(alto_2_fingering, alto_2_multiphonic[0])
2213
2214 fingering_6 = abjad.WoodwindFingering( #[1.5, 13.5, 22.5, 27, 30]
2215     name='alto-saxophone',
2216     center_column=('one', 'two', 'three', 'four', 'five', 'six', ),
2217     left_hand=(low-bes, ),
2218     right_hand=(),
2219 )
2220 command_6 = fingering_6()
2221 alto_3_fingering = abjad.Markup(contents=[size, command_6], direction=abjad.Down)
2222 alto_3_multiphonic = [x for x in abjad.iterate(score['Voice 7']).components(abjad.
Chord)]
2223 abjad.attach(alto_3_fingering, alto_3_multiphonic[0])
2224
2225 fingering_7 = abjad.WoodwindFingering( #[12.5, 15.5, 25.5]
2226     name='alto-saxophone',
2227     center_column=('one', 'two', 'three', 'four', 'five', ),
2228     left_hand=(b, ),
2229     right_hand=(low-c, c, ),
2230 )
2231 command_7 = fingering_7()
2232 alto_4_fingering = abjad.Markup(contents=[size, command_7], direction=abjad.Down)
2233 alto_4_multiphonic = [x for x in abjad.iterate(score['Voice 8']).components(abjad.
Chord)]
2234 abjad.attach(alto_4_fingering, alto_4_multiphonic[0])
2235
2236 fingering_8 = abjad.WoodwindFingering( #[1.5, 13.5, 22.5, 27, 30]
2237     name='alto-saxophone',
2238     center_column=('one', 'two', 'three', 'four', 'five', 'six', ),
2239     left_hand=(low-bes, ),
2240     right_hand=(),
2241 )
2242 command_8 = fingering_8()
2243 alto_5_fingering = abjad.Markup(contents=[size, command_8], direction=abjad.Down)

```

```

2244 alto_5_multiphonic = [x for x in abjad.iterate(score['Voice 9']).components(abjad.
2245     Chord)]
2246 abjad.attach(alto_5_fingering, alto_5_multiphonic[0])
2247
2247 fingering_9 = abjad.WoodwindFingering( #[6, 17.5]
2248     name='tenor-saxophone',
2249     center_column=( 'one', 'two', 'three', 'five', 'six', ),
2250     left_hand=( 'cis', ),
2251     right_hand=(),
2252 )
2253 command_9 = fingering_9()
2254 tenor_1_fingering = abjad.Markup(contents=[size, command_9], direction=abjad.Down)
2255 tenor_1_multiphonic = [x for x in abjad.iterate(score['Voice 11']).components(
2256     abjad.Chord)]
2256 abjad.attach(tenor_1_fingering, tenor_1_multiphonic[0])
2257 tenor_4_multiphonic = [x for x in abjad.iterate(score['Voice 14']).components(
2258     abjad.Chord)]
2259 abjad.attach(tenor_1_fingering, tenor_4_multiphonic[0])
2260
2260 fingering_10 = abjad.WoodwindFingering( #[6, 17.5, 25.5, 30]
2261     name='tenor-saxophone',
2262     center_column=( 'one', 'two', 'three', 'five', 'six', ),
2263     left_hand=( 'b', ),
2264     right_hand=(),
2265 )
2266 command_10 = fingering_10()
2267 tenor_2_fingering = abjad.Markup(contents=[size, command_10], direction=abjad.Down
2268 )
2268 tenor_2_multiphonic = [x for x in abjad.iterate(score['Voice 12']).components(
2269     abjad.Chord)]
2270 abjad.attach(tenor_2_fingering, tenor_2_multiphonic[0])
2271
2271 fingering_11 = abjad.WoodwindFingering( #[6, 17.5, 25.5, 30.5]
2272     name='tenor-saxophone',
2273     center_column=( 'one', 'two', 'three', 'five', 'six', ),
2274     left_hand=(),
2275     right_hand=(),
2276 )
2277 command_11 = fingering_11()
2278 tenor_3_fingering = abjad.Markup(contents=[size, command_11], direction=abjad.Down
2279 )
2279 tenor_3_multiphonic = [x for x in abjad.iterate(score['Voice 13']).components(
2280     abjad.Chord)]
2281 abjad.attach(tenor_3_fingering, tenor_3_multiphonic[0])
2282
2282 fingering_12 = abjad.WoodwindFingering( #[6, 17.5, 25.5, 30.5]
2283     name='tenor-saxophone',
2284     center_column=( 'one', 'two', 'three', 'five', 'six', ),
2285     left_hand=(),
2286     right_hand=(),
2287 )
2288 command_12 = fingering_12()
2289 tenor_5_fingering = abjad.Markup(contents=[size, command_12], direction=abjad.Down
2290 )
2290 tenor_5_multiphonic = [x for x in abjad.iterate(score['Voice 15']).components(
2291     abjad.Chord)]
2292 abjad.attach(tenor_5_fingering, tenor_5_multiphonic[0])
2293 fingering_13 = abjad.WoodwindFingering( #[13.5, 27.5, 33.5]

```

```

2294     name='baritone-saxophone',
2295     center_column=('one', 'two', 'three', ),
2296     left_hand=('ees', 'd', 'gis', ),
2297     right_hand=(),
2298 )
2299 command_13 = fingering_13()
2300 baritone_1_fingering = abjad.Markup(contents=[size, command_13], direction=abjad.
    Down)
2301 baritone_1_multiphonic = [x for x in abjad.iterate(score['Voice 16']).components(
    abjad.Chord)]
2302 abjad.attach(baritone_1_fingering, baritone_1_multiphonic[0])
2303
2304 fingering_14 = abjad.WoodwindFingering( #[4, 16.5, 23.5]
2305     name='baritone-saxophone',
2306     center_column=('one', 'two', 'three', 'four', 'five', ),
2307     left_hand=('b', ),
2308     right_hand=('low-c', ),
2309 )
2310 command_14 = fingering_14()
2311 baritone_2_fingering = abjad.Markup(contents=[size, command_14], direction=abjad.
    Down)
2312 baritone_2_multiphonic = [x for x in abjad.iterate(score['Voice 17']).components(
    abjad.Chord)]
2313 abjad.attach(baritone_2_fingering, baritone_2_multiphonic[0])
2314
2315 fingering_15 = abjad.WoodwindFingering( #[6.5, 17.5, 25.5, 34]
2316     name='baritone-saxophone',
2317     center_column=('one', 'two', 'three', 'five', 'six', ),
2318     left_hand=('low-a', ),
2319     right_hand=('low-c', ),
2320 )
2321 command_15 = fingering_15()
2322 baritone_3_fingering = abjad.Markup(contents=[size, command_15], direction=abjad.
    Down)
2323 baritone_3_multiphonic = [x for x in abjad.iterate(score['Voice 18']).components(
    abjad.Chord)]
2324 abjad.attach(baritone_3_fingering, baritone_3_multiphonic[0])
2325
2326 instruments = cyc([
2327     abjad.SopraninoSaxophone(),
2328     abjad.SopranoSaxophone(),
2329     abjad.SopranoSaxophone(),
2330     abjad.SopranoSaxophone(),
2331     abjad.AltoSaxophone(),
2332     abjad.AltoSaxophone(),
2333     abjad.AltoSaxophone(),
2334     abjad.AltoSaxophone(),
2335     abjad.AltoSaxophone(),
2336     abjad.AltoSaxophone(),
2337     abjad.TenorSaxophone(),
2338     abjad.TenorSaxophone(),
2339     abjad.TenorSaxophone(),
2340     abjad.TenorSaxophone(),
2341     abjad.TenorSaxophone(),
2342     abjad.BaritoneSaxophone(),
2343     abjad.BaritoneSaxophone(),
2344     abjad.BaritoneSaxophone(),
2345     abjad.BassSaxophone(),
2346     abjad.BassSaxophone(),

```

```

2347     abjad.ContrabassSaxophone(),
2348 ])
2349
2350 abbreviations = cyc([
2351     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
2352     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
2353     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
2354     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
2355     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
2356     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
2357     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
2358     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
2359     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
2360     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
2361     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
2362     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
2363     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
2364     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
2365     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
2366     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
2367     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
2368     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
2369     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
2370     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
2371     abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
2372 ])
2373
2374 names = cyc([
2375     abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
2376     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
2377     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
2378     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
2379     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
2380     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
2381     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
2382     abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
2383     abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
2384     abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
2385     abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
2386     abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
2387     abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
2388     abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
2389     abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
2390     abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
2391     abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
2392     abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
2393     abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
2394     abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
2395     abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
2396 ])
2397
2398 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2399     leaf1 = abjad.select(staff).leaves()[0]
2400     abjad.attach(next(instruments), leaf1)
2401     abjad.attach(next(abbreviations), leaf1)
2402     abjad.attach(next(names), leaf1)
2403
2404 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
2405     leaf1 = abjad.select(staff).leaves()[0]

```

```

2406 last_leaf = abjad.select(staff).leaves()[-1]
2407 # abjad.attach(metro, leaf1)
2408 abjad.attach(bar_line, last_leaf)
2409
2410 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
2411     leaf1 = abjad.select(staff).leaves()[0]
2412     abjad.attach(mark, leaf1)
2413
2414 score_file = abjad.LilyPondFile.new(
2415     score,
2416     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2417     _stylesheets/abjad.ily'],
2418 )
2419
2420 abjad.SegmentMaker.comment_measure_numbers(score)
2421 ######
2422 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_A'
2423 pdf_path = f'{directory}/Section_A.pdf'
2424 path = pathlib.Path('Section_A.pdf')
2425 if path.exists():
2426     print(f'Removing {pdf_path} ...')
2427     path.unlink()
2428 time_1 = time.time()
2429 print(f'Persisting {pdf_path} ...')
2430 result = abjad.persist(score_file).as_pdf(pdf_path)
2431 print(result[0])
2432 print(result[1])
2433 print(result[2])
2434 success = result[3]
2435 if success is False:
2436     print('LilyPond failed!')
2437 time_2 = time.time()
2438 total_time = time_2 - time_1
2439 print(f'Total time: {total_time} seconds')
2440 if path.exists():
2441     print(f'Opening {pdf_path} ...')
2442     os.system(f'open {pdf_path}')
2443 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_A/Section_A.
2444 ly').readlines()
2445 open('/Users/evansdsg2/Scores/guerrero/Build/Section_A.ly', 'w').writelines(
2446     score_lines[15:-1])
2447
2448 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
2449     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2450     signature_copy = abjad.mutate(signatures).copy()
2451     staff_copy = abjad.mutate(staff).copy()
2452     part = abjad.Score()
2453     part.insert(0, staff)
2454     part.insert(0, signature_copy)
2455     part_file = abjad.LilyPondFile.new(
2456         part,
2457         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2458         _stylesheets/abjad.ily'],
2459     )
2460     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.) soprano'
2461     pdf_path = f'{directory}/Section_A.pdf'
2462     path = pathlib.Path('Section_A.pdf')
2463     if path.exists():

```

```

2461     print(f'Removing {pdf_path} ...')
2462     path.unlink()
2463 time_1 = time.time()
2464 print(f'Persisting {pdf_path} ...')
2465 result = abjad.persist(part_file).as_pdf(pdf_path)
2466 print(result[0])
2467 print(result[1])
2468 print(result[2])
2469 success = result[3]
2470 if success is False:
2471     print('LilyPond failed!')
2472 time_2 = time.time()
2473 total_time = time_2 - time_1
2474 print(f'Total time: {total_time} seconds')
2475 if path.exists():
2476     print(f'Opening {pdf_path} ...')
2477     os.system(f'open {pdf_path}')
2478 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano1/
Section_A.ly').readlines()
2479 open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano1/Section_A.ly',
'w').writelines(part_lines[15:-1])
2480
2481 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
2482 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2483 signature_copy = abjad.mutate(signatures).copy()
2484 staff_copy = abjad.mutate(staff).copy()
2485 part = abjad.Score()
2486 part.insert(0, staff)
2487 part.insert(0, signature_copy)
2488 part_file = abjad.LilyPondFile.new(
2489     part,
2490     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2491     )
2492 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1'
2493 pdf_path = f'{directory}/Section_A.pdf'
2494 path = pathlib.Path('Section_A.pdf')
2495 if path.exists():
2496     print(f'Removing {pdf_path} ...')
2497     path.unlink()
2498 time_1 = time.time()
2499 print(f'Persisting {pdf_path} ...')
2500 result = abjad.persist(part_file).as_pdf(pdf_path)
2501 print(result[0])
2502 print(result[1])
2503 print(result[2])
2504 success = result[3]
2505 if success is False:
2506     print('LilyPond failed!')
2507 time_2 = time.time()
2508 total_time = time_2 - time_1
2509 print(f'Total time: {total_time} seconds')
2510 if path.exists():
2511     print(f'Opening {pdf_path} ...')
2512     os.system(f'open {pdf_path}')
2513 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/
Section_A.ly').readlines()
2514 open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/Section_A.ly',
'w').writelines(part_lines[15:-1])

```

```

2515
2516 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
2517     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2518     signature_copy = abjad.mutate(signatures).copy()
2519     staff_copy = abjad.mutate(staff).copy()
2520     part = abjad.Score()
2521     part.insert(0, staff)
2522     part.insert(0, signature_copy)
2523     part_file = abjad.LilyPondFile.new(
2524         part,
2525         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2526         _stylesheets/abjad.ily'],
2527         )
2528     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
2529     pdf_path = f'{directory}/Section_A.pdf'
2530     path = pathlib.Path('Section_A.pdf')
2531     if path.exists():
2532         print(f'Removing {pdf_path} ...')
2533         path.unlink()
2534     time_1 = time.time()
2535     print(f'Persisting {pdf_path} ...')
2536     result = abjad.persist(part_file).as_pdf(pdf_path)
2537     print(result[0])
2538     print(result[1])
2539     print(result[2])
2540     success = result[3]
2541     if success is False:
2542         print('LilyPond failed!')
2543     time_2 = time.time()
2544     total_time = time_2 - time_1
2545     print(f'Total time: {total_time} seconds')
2546     if path.exists():
2547         print(f'Opening {pdf_path} ...')
2548         os.system(f'open {pdf_path}')
2549     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/
2550     Section_A.ly').readlines()
2551     open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/Section_A.ly',
2552     'w').writelines(part_lines[15:-1])
2553
2554 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
2555     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2556     signature_copy = abjad.mutate(signatures).copy()
2557     staff_copy = abjad.mutate(staff).copy()
2558     part = abjad.Score()
2559     part.insert(0, staff)
2560     part.insert(0, signature_copy)
2561     part_file = abjad.LilyPondFile.new(
2562         part,
2563         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2564         _stylesheets/abjad.ily'],
2565         )
2566     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3'
2567     pdf_path = f'{directory}/Section_A.pdf'
2568     path = pathlib.Path('Section_A.pdf')
2569     if path.exists():
2570         print(f'Removing {pdf_path} ...')
2571         path.unlink()
2572     time_1 = time.time()
2573     print(f'Persisting {pdf_path} ...')

```

```

2570     result = abjad.persist(part_file).as_pdf(pdf_path)
2571     print(result[0])
2572     print(result[1])
2573     print(result[2])
2574     success = result[3]
2575     if success is False:
2576         print('LilyPond failed!')
2577     time_2 = time.time()
2578     total_time = time_2 - time_1
2579     print(f'Total time: {total_time} seconds')
2580     if path.exists():
2581         print(f'Opening {pdf_path} ...')
2582         os.system(f'open {pdf_path}')
2583     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/
2584     Section_A.ly').readlines()
2585     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/Section_A.ly',
2586     'w').writelines(part_lines[15:-1])
2587
2588 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
2589     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2590     signature_copy = abjad.mutate(signatures).copy()
2591     staff_copy = abjad.mutate(staff).copy()
2592     part = abjad.Score()
2593     part.insert(0, staff)
2594     part.insert(0, signature_copy)
2595     part_file = abjad.LilyPondFile.new(
2596         part,
2597         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2598         _stylesheets/abjad.ily'],
2599         )
2600     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1'
2601     pdf_path = f'{directory}/Section_A.pdf'
2602     path = pathlib.Path('Section_A.pdf')
2603     if path.exists():
2604         print(f'Removing {pdf_path} ...')
2605         path.unlink()
2606     time_1 = time.time()
2607     print(f'Persisting {pdf_path} ...')
2608     result = abjad.persist(part_file).as_pdf(pdf_path)
2609     print(result[0])
2610     print(result[1])
2611     print(result[2])
2612     success = result[3]
2613     if success is False:
2614         print('LilyPond failed!')
2615     time_2 = time.time()
2616     total_time = time_2 - time_1
2617     print(f'Total time: {total_time} seconds')
2618     if path.exists():
2619         print(f'Opening {pdf_path} ...')
2620         os.system(f'open {pdf_path}')
2621     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/
2622     Section_A.ly').readlines()
2623     open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/Section_A.ly',
2624     'w').writelines(part_lines[15:-1])
2625
2626 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
2627     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2628     signature_copy = abjad.mutate(signatures).copy()

```

```

2624 staff_copy = abjad.mutate(staff).copy()
2625 part = abjad.Score()
2626 part.insert(0, staff)
2627 part.insert(0, signature_copy)
2628 part_file = abjad.LilyPondFile.new(
2629     part,
2630     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2631 _stylesheets/abjad.ily'],
2632 )
2633 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
2634 pdf_path = f'{directory}/Section_A.pdf'
2635 path = pathlib.Path('Section_A.pdf')
2636 if path.exists():
2637     print(f'Removing {pdf_path} ...')
2638     path.unlink()
2639 time_1 = time.time()
2640 print(f'Persisting {pdf_path} ...')
2641 result = abjad.persist(part_file).as_pdf(pdf_path)
2642 print(result[0])
2643 print(result[1])
2644 print(result[2])
2645 success = result[3]
2646 if success is False:
2647     print('LilyPond failed!')
2648 time_2 = time.time()
2649 total_time = time_2 - time_1
2650 print(f'Total time: {total_time} seconds')
2651 if path.exists():
2652     print(f'Opening {pdf_path} ...')
2653     os.system(f'open {pdf_path}')
2654 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
2655 Section_A.ly').readlines()
2656 open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_A.ly', 'w'
2657 .writelines(part_lines[15:-1])
2658
2659 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
2660     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2661     signature_copy = abjad.mutate(signatures).copy()
2662     staff_copy = abjad.mutate(staff).copy()
2663     part = abjad.Score()
2664     part.insert(0, staff)
2665     part.insert(0, signature_copy)
2666     part_file = abjad.LilyPondFile.new(
2667         part,
2668         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2669 _stylesheets/abjad.ily'],
2670     )
2671     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
2672     pdf_path = f'{directory}/Section_A.pdf'
2673     path = pathlib.Path('Section_A.pdf')
2674     if path.exists():
2675         print(f'Removing {pdf_path} ...')
2676         path.unlink()
2677     time_1 = time.time()
2678     print(f'Persisting {pdf_path} ...')
2679     result = abjad.persist(part_file).as_pdf(pdf_path)
2680     print(result[0])
2681     print(result[1])
2682     print(result[2])

```

```

2679     success = result[3]
2680     if success is False:
2681         print('LilyPond failed!')
2682     time_2 = time.time()
2683     total_time = time_2 - time_1
2684     print(f'Total time: {total_time} seconds')
2685     if path.exists():
2686         print(f'Opening {pdf_path} ...')
2687         os.system(f'open {pdf_path}')
2688         part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.')alto3/
2689         Section_A.ly').readlines()
2690         open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.')alto3/Section_A.ly', 'w'
2691         ).writelines(part_lines[15:-1])
2692
2693 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
2694     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2695     signature_copy = abjad.mutate(signatures).copy()
2696     staff_copy = abjad.mutate(staff).copy()
2697     part = abjad.Score()
2698     part.insert(0, staff)
2699     part.insert(0, signature_copy)
2700     part_file = abjad.LilyPondFile.new(
2701         part,
2702         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2703         _stylesheets/abjad.ily'],
2704         )
2705     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4'
2706     pdf_path = f'{directory}/Section_A.pdf'
2707     path = pathlib.Path('Section_A.pdf')
2708     if path.exists():
2709         print(f'Removing {pdf_path} ...')
2710         path.unlink()
2711     time_1 = time.time()
2712     print(f'Persisting {pdf_path} ...')
2713     result = abjad.persist(part_file).as_pdf(pdf_path)
2714     print(result[0])
2715     print(result[1])
2716     print(result[2])
2717     success = result[3]
2718     if success is False:
2719         print('LilyPond failed!')
2720     time_2 = time.time()
2721     total_time = time_2 - time_1
2722     print(f'Total time: {total_time} seconds')
2723     if path.exists():
2724         print(f'Opening {pdf_path} ...')
2725         os.system(f'open {pdf_path}')
2726         part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/
2727         Section_A.ly').readlines()
2728         open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/Section_A.ly', 'w'
2729         ).writelines(part_lines[15:-1])
2730
2731 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
2732     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2733     signature_copy = abjad.mutate(signatures).copy()
2734     staff_copy = abjad.mutate(staff).copy()
2735     part = abjad.Score()
2736     part.insert(0, staff)
2737     part.insert(0, signature_copy)

```

```
2733 part_file = abjad.LilyPondFile.new(
2734     part,
2735     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2736 _stylesheets/abjad.ily'],
2737     )
2738 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
2739 pdf_path = f'{directory}/Section_A.pdf'
2740 path = pathlib.Path('Section_A.pdf')
2741 if path.exists():
2742     print(f'Removing {pdf_path} ...')
2743     path.unlink()
2744 time_1 = time.time()
2745 print(f'Persisting {pdf_path} ...')
2746 result = abjad.persist(part_file).as_pdf(pdf_path)
2747 print(result[0])
2748 print(result[1])
2749 print(result[2])
2750 success = result[3]
2751 if success is False:
2752     print('LilyPond failed!')
2753 time_2 = time.time()
2754 total_time = time_2 - time_1
2755 print(f'Total time: {total_time} seconds')
2756 if path.exists():
2757     print(f'Opening {pdf_path} ...')
2758     os.system(f'open {pdf_path}')
2759 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/
2760 Section_A.ly').readlines()
2761 open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/Section_A.ly', 'w'
2762 .writelines(part_lines[15:-1])
2763
2764 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
2765     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2766     signature_copy = abjad.mutate(signatures).copy()
2767     staff_copy = abjad.mutate(staff).copy()
2768     part = abjad.Score()
2769     part.insert(0, staff)
2770     part.insert(0, signature_copy)
2771     part_file = abjad.LilyPondFile.new(
2772         part,
2773         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2774 _stylesheets/abjad.ily'],
2775         )
2776 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
2777 pdf_path = f'{directory}/Section_A.pdf'
2778 path = pathlib.Path('Section_A.pdf')
2779 if path.exists():
2780     print(f'Removing {pdf_path} ...')
2781     path.unlink()
2782 time_1 = time.time()
2783 print(f'Persisting {pdf_path} ...')
2784 result = abjad.persist(part_file).as_pdf(pdf_path)
2785 print(result[0])
2786 print(result[1])
2787 print(result[2])
2788 success = result[3]
2789 if success is False:
2790     print('LilyPond failed!')
2791 time_2 = time.time()
```

```

2788     total_time = time_2 - time_1
2789     print(f'Total time: {total_time} seconds')
2790     if path.exists():
2791         print(f'Opening {pdf_path} ...')
2792         os.system(f'open {pdf_path}')
2793     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
2794     Section_A.ly').readlines()
2795     open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_A.ly', 'w'
2796     ).writelines(part_lines[15:-1])
2797
2798 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
2799     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2800     signature_copy = abjad.mutate(signatures).copy()
2801     staff_copy = abjad.mutate(staff).copy()
2802     part = abjad.Score()
2803     part.insert(0, staff)
2804     part.insert(0, signature_copy)
2805     part_file = abjad.LilyPondFile.new(
2806         part,
2807         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2808         _stylesheets/abjad.ily'],
2809         )
2810     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenori'
2811     pdf_path = f'{directory}/Section_A.pdf'
2812     path = pathlib.Path('Section_A.pdf')
2813     if path.exists():
2814         print(f'Removing {pdf_path} ...')
2815         path.unlink()
2816     time_1 = time.time()
2817     print(f'Persisting {pdf_path} ...')
2818     result = abjad.persist(part_file).as_pdf(pdf_path)
2819     print(result[0])
2820     print(result[1])
2821     print(result[2])
2822     success = result[3]
2823     if success is False:
2824         print('LilyPond failed!')
2825     time_2 = time.time()
2826     total_time = time_2 - time_1
2827     print(f'Total time: {total_time} seconds')
2828     if path.exists():
2829         print(f'Opening {pdf_path} ...')
2830         os.system(f'open {pdf_path}')
2831     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenori/
2832     Section_A.ly').readlines()
2833     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenori/Section_A.ly', 'w'
2834     ).writelines(part_lines[15:-1])
2835
2836 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
2837     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2838     signature_copy = abjad.mutate(signatures).copy()
2839     staff_copy = abjad.mutate(staff).copy()
2840     part = abjad.Score()
2841     part.insert(0, staff)
2842     part.insert(0, signature_copy)
2843     part_file = abjad.LilyPondFile.new(
2844         part,
2845         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2846         _stylesheets/abjad.ily'],

```

```

2841     )
2842     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
2843     pdf_path = f'{directory}/Section_A.pdf'
2844     path = pathlib.Path('Section_A.pdf')
2845     if path.exists():
2846         print(f'Removing {pdf_path} ...')
2847         path.unlink()
2848     time_1 = time.time()
2849     print(f'Persisting {pdf_path} ...')
2850     result = abjad.persist(part_file).as_pdf(pdf_path)
2851     print(result[0])
2852     print(result[1])
2853     print(result[2])
2854     success = result[3]
2855     if success is False:
2856         print('LilyPond failed!')
2857     time_2 = time.time()
2858     total_time = time_2 - time_1
2859     print(f'Total time: {total_time} seconds')
2860     if path.exists():
2861         print(f'Opening {pdf_path} ...')
2862         os.system(f'open {pdf_path}')
2863     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/
2864     Section_A.ly').readlines()
2865     open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/Section_A.ly', 'w').writelines(part_lines[15:-1])
2866
2867 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
2868     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2869     signature_copy = abjad.mutate(signatures).copy()
2870     staff_copy = abjad.mutate(staff).copy()
2871     part = abjad.Score()
2872     part.insert(0, staff)
2873     part.insert(0, signature_copy)
2874     part_file = abjad.LilyPondFile.new(
2875         part,
2876         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2877         _stylesheets/abjad.ily'],
2878         )
2879     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3'
2880     pdf_path = f'{directory}/Section_A.pdf'
2881     path = pathlib.Path('Section_A.pdf')
2882     if path.exists():
2883         print(f'Removing {pdf_path} ...')
2884         path.unlink()
2885     time_1 = time.time()
2886     print(f'Persisting {pdf_path} ...')
2887     result = abjad.persist(part_file).as_pdf(pdf_path)
2888     print(result[0])
2889     print(result[1])
2890     print(result[2])
2891     success = result[3]
2892     if success is False:
2893         print('LilyPond failed!')
2894     time_2 = time.time()
2895     total_time = time_2 - time_1
2896     print(f'Total time: {total_time} seconds')
2897     if path.exists():
2898         print(f'Opening {pdf_path} ...')

```

```

2897     os.system(f'open {pdf_path}')
2898     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
2899     Section_A.ly').readlines()
2900     open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_A.ly', ,
2901     w').writelines(part_lines[15:-1])
2900
2901 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
2902     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2903     signature_copy = abjad.mutate(signatures).copy()
2904     staff_copy = abjad.mutate(staff).copy()
2905     part = abjad.Score()
2906     part.insert(0, staff)
2907     part.insert(0, signature_copy)
2908     part_file = abjad.LilyPondFile.new(
2909         part,
2910         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2911         _stylesheets/abjad.ily'],
2912         )
2913     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'
2914     pdf_path = f'{directory}/Section_A.pdf'
2915     path = pathlib.Path('Section_A.pdf')
2916     if path.exists():
2917         print(f'Removing {pdf_path} ...')
2918         path.unlink()
2919     time_1 = time.time()
2920     print(f'Persisting {pdf_path} ...')
2921     result = abjad.persist(part_file).as_pdf(pdf_path)
2922     print(result[0])
2923     print(result[1])
2924     print(result[2])
2925     success = result[3]
2926     if success is False:
2927         print('LilyPond failed!')
2928     time_2 = time.time()
2929     total_time = time_2 - time_1
2930     print(f'Total time: {total_time} seconds')
2931     if path.exists():
2932         print(f'Opening {pdf_path} ...')
2933         os.system(f'open {pdf_path}')
2934     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
2935     Section_A.ly').readlines()
2936     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_A.ly', ,
2937     w').writelines(part_lines[15:-1])
2938
2939 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
2940     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2941     signature_copy = abjad.mutate(signatures).copy()
2942     staff_copy = abjad.mutate(staff).copy()
2943     part = abjad.Score()
2944     part.insert(0, staff)
2945     part.insert(0, signature_copy)
2946     part_file = abjad.LilyPondFile.new(
2947         part,
2948         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2949         _stylesheets/abjad.ily'],
2950         )
2951     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5'
2952     pdf_path = f'{directory}/Section_A.pdf'
2953     path = pathlib.Path('Section_A.pdf')

```

```

2950     if path.exists():
2951         print(f'Removing {pdf_path} ...')
2952         path.unlink()
2953     time_1 = time.time()
2954     print(f'Persisting {pdf_path} ...')
2955     result = abjad.persist(part_file).as_pdf(pdf_path)
2956     print(result[0])
2957     print(result[1])
2958     print(result[2])
2959     success = result[3]
2960     if success is False:
2961         print('LilyPond failed!')
2962     time_2 = time.time()
2963     total_time = time_2 - time_1
2964     print(f'Total time: {total_time} seconds')
2965     if path.exists():
2966         print(f'Opening {pdf_path} ...')
2967         os.system(f'open {pdf_path}')
2968     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/
Section_A.ly').readlines()
2969     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/Section_A.ly', 'w').writelines(part_lines[15:-1])
2970
2971 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
2972     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2973     signature_copy = abjad.mutate(signatures).copy()
2974     staff_copy = abjad.mutate(staff).copy()
2975     part = abjad.Score()
2976     part.insert(0, staff)
2977     part.insert(0, signature_copy)
2978     part_file = abjad.LilyPondFile.new(
2979         part,
2980         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2981         )
2982     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1'
2983     pdf_path = f'{directory}/Section_A.pdf'
2984     path = pathlib.Path('Section_A.pdf')
2985     if path.exists():
2986         print(f'Removing {pdf_path} ...')
2987         path.unlink()
2988     time_1 = time.time()
2989     print(f'Persisting {pdf_path} ...')
2990     result = abjad.persist(part_file).as_pdf(pdf_path)
2991     print(result[0])
2992     print(result[1])
2993     print(result[2])
2994     success = result[3]
2995     if success is False:
2996         print('LilyPond failed!')
2997     time_2 = time.time()
2998     total_time = time_2 - time_1
2999     print(f'Total time: {total_time} seconds')
3000     if path.exists():
3001         print(f'Opening {pdf_path} ...')
3002         os.system(f'open {pdf_path}')
3003     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/
Section_A.ly').readlines()
3004     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/Section_A.ly',

```

```

3005     , 'w').writelines(part_lines[15:-1])
3006
3006 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
3007     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3008     signature_copy = abjad.mutate(signatures).copy()
3009     staff_copy = abjad.mutate(staff).copy()
3010     part = abjad.Score()
3011     part.insert(0, staff)
3012     part.insert(0, signature_copy)
3013     part_file = abjad.LilyPondFile.new(
3014         part,
3015         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3016         _stylesheets/abjad.ily'],
3017         )
3017     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2'
3018     pdf_path = f'{directory}/Section_A.pdf'
3019     path = pathlib.Path('Section_A.pdf')
3020     if path.exists():
3021         print(f'Removing {pdf_path} ...')
3022         path.unlink()
3023     time_1 = time.time()
3024     print(f'Persisting {pdf_path} ...')
3025     result = abjad.persist(part_file).as_pdf(pdf_path)
3026     print(result[0])
3027     print(result[1])
3028     print(result[2])
3029     success = result[3]
3030     if success is False:
3031         print('LilyPond failed!')
3032     time_2 = time.time()
3033     total_time = time_2 - time_1
3034     print(f'Total time: {total_time} seconds')
3035     if path.exists():
3036         print(f'Opening {pdf_path} ...')
3037         os.system(f'open {pdf_path}')
3038     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/
3039     Section_A.ly').readlines()
3040     open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/Section_A.ly',
3041     'w').writelines(part_lines[15:-1])
3042
3041 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
3042     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3043     signature_copy = abjad.mutate(signatures).copy()
3044     staff_copy = abjad.mutate(staff).copy()
3045     part = abjad.Score()
3046     part.insert(0, staff)
3047     part.insert(0, signature_copy)
3048     part_file = abjad.LilyPondFile.new(
3049         part,
3050         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3051         _stylesheets/abjad.ily'],
3052         )
3052     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3'
3053     pdf_path = f'{directory}/Section_A.pdf'
3054     path = pathlib.Path('Section_A.pdf')
3055     if path.exists():
3056         print(f'Removing {pdf_path} ...')
3057         path.unlink()
3058     time_1 = time.time()

```

```

3059     print(f'Persisting {pdf_path} ...')
3060     result = abjad.persist(part_file).as_pdf(pdf_path)
3061     print(result[0])
3062     print(result[1])
3063     print(result[2])
3064     success = result[3]
3065     if success is False:
3066         print('LilyPond failed!')
3067     time_2 = time.time()
3068     total_time = time_2 - time_1
3069     print(f'Total time: {total_time} seconds')
3070     if path.exists():
3071         print(f'Opening {pdf_path} ...')
3072         os.system(f'open {pdf_path}')
3073     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/
3074     Section_A.ly').readlines()
3075     open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/Section_A.ly',
3076     , 'w').writelines(part_lines[15:-1])
3077
3078 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
3079     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3080     signature_copy = abjad.mutate(signatures).copy()
3081     staff_copy = abjad.mutate(staff).copy()
3082     part = abjad.Score()
3083     part.insert(0, staff)
3084     part.insert(0, signature_copy)
3085     part_file = abjad.LilyPondFile.new(
3086         part,
3087         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3088         _stylesheets/abjad.ily'],
3089         )
3090     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.')bass1'
3091     pdf_path = f'{directory}/Section_A.pdf'
3092     path = pathlib.Path('Section_A.pdf')
3093     if path.exists():
3094         print(f'Removing {pdf_path} ...')
3095         path.unlink()
3096     time_1 = time.time()
3097     print(f'Persisting {pdf_path} ...')
3098     result = abjad.persist(part_file).as_pdf(pdf_path)
3099     print(result[0])
3100     print(result[1])
3101     print(result[2])
3102     success = result[3]
3103     if success is False:
3104         print('LilyPond failed!')
3105     time_2 = time.time()
3106     total_time = time_2 - time_1
3107     print(f'Total time: {total_time} seconds')
3108     if path.exists():
3109         print(f'Opening {pdf_path} ...')
3110         os.system(f'open {pdf_path}')
3111     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.')bass1/
3112     Section_A.ly').readlines()
3113     open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.')bass1/Section_A.ly',
3114     , 'w').writelines(part_lines[15:-1])
3115
3116 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
3117     signatures = abjad.select(score['Global Context']).components(abjad.Staff)

```

```

3113 signature_copy = abjad.mutate(signatures).copy()
3114 staff_copy = abjad.mutate(staff).copy()
3115 part = abjad.Score()
3116 part.insert(0, staff)
3117 part.insert(0, signature_copy)
3118 part_file = abjad.LilyPondFile.new(
3119     part,
3120     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3121 _stylesheets/abjad.ly'],
3122     )
3123 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
3124 pdf_path = f'{directory}/Section_A.pdf'
3125 path = pathlib.Path('Section_A.pdf')
3126 if path.exists():
3127     print(f'Removing {pdf_path} ...')
3128     path.unlink()
3129 time_1 = time.time()
3130 print(f'Persisting {pdf_path} ...')
3131 result = abjad.persist(part_file).as_pdf(pdf_path)
3132 print(result[0])
3133 print(result[1])
3134 print(result[2])
3135 success = result[3]
3136 if success is False:
3137     print('LilyPond failed!')
3138 time_2 = time.time()
3139 total_time = time_2 - time_1
3140 print(f'Total time: {total_time} seconds')
3141 if path.exists():
3142     print(f'Opening {pdf_path} ...')
3143     os.system(f'open {pdf_path}')
3144 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
3145 Section_A.ly').readlines()
3146 open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_A.ly', 'w'
3147 .writelines(part_lines[15:-1])
3148
3149 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
3150     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3151     signature_copy = abjad.mutate(signatures).copy()
3152     staff_copy = abjad.mutate(staff).copy()
3153     part = abjad.Score()
3154     part.insert(0, staff)
3155     part.insert(0, signature_copy)
3156     part_file = abjad.LilyPondFile.new(
3157         part,
3158         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3159 _stylesheets/abjad.ly'],
3160         )
3161     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
3162     pdf_path = f'{directory}/Section_A.pdf'
3163     path = pathlib.Path('Section_A.pdf')
3164     if path.exists():
3165         print(f'Removing {pdf_path} ...')
3166         path.unlink()
3167     time_1 = time.time()
3168     print(f'Persisting {pdf_path} ...')
3169     result = abjad.persist(part_file).as_pdf(pdf_path)
3170     print(result[0])
3171     print(result[1])

```

```

3168     print(result[2])
3169     success = result[3]
3170     if success is False:
3171         print('LilyPond failed!')
3172     time_2 = time.time()
3173     total_time = time_2 - time_1
3174     print(f'Total time: {total_time} seconds')
3175     if path.exists():
3176         print(f'Opening {pdf_path} ...')
3177         os.system(f'open {pdf_path}')
3178     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass
/Section_A.ly').readlines()
3179     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_A.ly
', 'w').writelines(part_lines[15:-1])

```

Listing 3.2: Invocation Source Code

### 3.3 Section B

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11 from evans.general_tools.random_walk import randomWalk
12 from evans.general_tools.rotate import rotate
13 from evans.general_tools.mirror import mirror
14
15 print('Interpreting file ...')
16
17 time_signatures = [
18     abjad.TimeSignature(pair) for pair in [
19         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
20         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
21         (4, 4), (4, 4), (4, 4),
22     ]
23 ]
24
25 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
26
27 def cyc(lst):
28     count = 0
29     while True:
30         yield lst[count%len(lst)]
31         count += 1
32
33 def grouper(lst1, lst2):
34     def cyc(lst):
35         c = 0
36         while True:
37             yield lst[c%len(lst)]
38             c += 1

```

```

39     lst1 = cyc(lst1)
40     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2
41     ]
42
43 def reduceMod(list_length, rw):
44     return [(x % list_length) for x in rw]
45
46 # -3 at bottom of chord for completion
47 soprano_chord = [27, ]
48 soprano_1_chord = [[13.25, 16, 26.25, ], 22,]
49 soprano_2_chord = [[13, 14.75, 26.25, ], 16,] #maybe it's 13.25?
50 soprano_3_chord = [[12.75, 15.5, 26, ], 13,]
51 alto_1_chord = [[12.5, 19, 27.75, 34, ], 20,]
52 alto_2_chord = [[12.5, 15.25, 25.5, ], 12,]
53 alto_3_chord = [[1.75, 13.5, 22.25, 27, 30, ], 1,]
54 alto_4_chord = [[12.5, 15.25, 25.5, ], 20,]
55 alto_5_chord = [[1.75, 13.5, 22.25, 27, 30, ], 12,]
56 alto_6_chord = [[12.5, 19, 27.75, 34, ], 1,]
57 tenor_1_chord = [[6, 17.5, ], 17,]
58 tenor_2_chord = [[6, 17.5, 25.5, 30, ], 6,]
59 tenor_3_chord = [[6, 17.5, 25.5, 30.75, ], -1]
60 tenor_4_chord = [[6, 17.5, ], 17,]
61 tenor_5_chord = [[6, 17.5, 25.5, 30.75, ], 6,]
62 baritone_1_chord = [[13.25, 27.5, 33.75, ], 13,]
63 baritone_2_chord = [[4, 16.5, 23.5, ], 6,]
64 baritone_3_chord = [[7.75, 17.75, 25.5, 34, ], 4,]
65 bass_1_chord = [11, ]
66 bass_2_chord = [9, ]
67 contrabass_chord = [-2, 2, 7, -2, 2, 7, 2, -2]
68
69 def reduceMod(x, rw):
70     return [(y % x) for y in rw]
71
72 walk_list = []
73 for x in range(-1, 30):
74     walk_list.append(x)
75     walk_list.append(x + 0.5)
76 mirrored_walk_list = mirror(walk_list, sequential_duplicates=False)
77
78 soprano_walk_chord = rotate(mirrored_walk_list, 56)
79 soprano_random_walk_notes = [x for x in randomWalk(
80     random_seed=1,
81     length=1000,
82     step_list=[1],
83     mapped_list=soprano_walk_chord
84     )
85     ]
86
87 soprano_1_walk_chord = rotate(mirrored_walk_list, 46)
88 soprano_1_random_walk_notes = [x for x in randomWalk(
89     random_seed=2,
90     length=1000,
91     step_list=[1],
92     mapped_list=soprano_1_walk_chord
93     )
94     ]
95
96 soprano_2_walk_chord = rotate(mirrored_walk_list, 34)

```

```
97 soprano_2_random_walk_notes = [x for x in randomWalk(
98     random_seed=3,
99     length=1000,
100    step_list=[1],
101    mapped_list=soprano_2_walk_chord
102        )
103    ]
104
105 soprano_3_walk_chord = rotate(mirrored_walk_list, 28)
106 soprano_3_random_walk_notes = [x for x in randomWalk(
107     random_seed=4,
108     length=1000,
109    step_list=[1],
110    mapped_list=soprano_3_walk_chord
111        )
112    ]
113
114 alto_1_walk_chord = rotate(mirrored_walk_list, 42)
115 alto_1_random_walk_notes = [x for x in randomWalk(
116     random_seed=5,
117     length=1000,
118    step_list=[1],
119    mapped_list=alto_1_walk_chord
120        )
121    ]
122
123 alto_2_walk_chord = rotate(mirrored_walk_list, 26)
124 alto_2_random_walk_notes = [x for x in randomWalk(
125     random_seed=6,
126     length=1000,
127    step_list=[1],
128    mapped_list=alto_2_walk_chord
129        )
130    ]
131
132 alto_3_walk_chord = rotate(mirrored_walk_list, 4)
133 alto_3_random_walk_notes = [x for x in randomWalk(
134     random_seed=7,
135     length=1000,
136    step_list=[1],
137    mapped_list=alto_3_walk_chord
138        )
139    ]
140
141 alto_4_walk_chord = rotate(mirrored_walk_list, 42)
142 alto_4_random_walk_notes = [x for x in randomWalk(
143     random_seed=8,
144     length=1000,
145    step_list=[1],
146    mapped_list=alto_4_walk_chord
147        )
148    ]
149
150 alto_5_walk_chord = rotate(mirrored_walk_list, 26)
151 alto_5_random_walk_notes = [x for x in randomWalk(
152     random_seed=9,
153     length=1000,
154    step_list=[1],
155    mapped_list=alto_5_walk_chord
```

```
156         )
157     ]
158
159 alto_6_walk_chord = rotate(mirrored_walk_list, 4)
160 alto_6_random_walk_notes = [x for x in randomWalk(
161     random_seed=10,
162     length=1000,
163     step_list=[1],
164     mapped_list=alto_6_walk_chord
165     )
166   ]
167
168 tenor_1_walk_chord = rotate(mirrored_walk_list, 36)
169 tenor_1_random_walk_notes = [x for x in randomWalk(
170     random_seed=11,
171     length=1000,
172     step_list=[1],
173     mapped_list=tenor_1_walk_chord
174     )
175   ]
176
177 tenor_2_walk_chord = rotate(mirrored_walk_list, 14)
178 tenor_2_random_walk_notes = [x for x in randomWalk(
179     random_seed=12,
180     length=1000,
181     step_list=[1],
182     mapped_list=tenor_2_walk_chord
183     )
184   ]
185
186 tenor_3_walk_chord = rotate(mirrored_walk_list, 0)
187 tenor_3_random_walk_notes = [x for x in randomWalk(
188     random_seed=13,
189     length=1000,
190     step_list=[1],
191     mapped_list=tenor_3_walk_chord
192     )
193   ]
194
195 tenor_4_walk_chord = rotate(mirrored_walk_list, 36)
196 tenor_4_random_walk_notes = [x for x in randomWalk(
197     random_seed=14,
198     length=1000,
199     step_list=[1],
200     mapped_list=tenor_4_walk_chord
201     )
202   ]
203
204 tenor_5_walk_chord = rotate(mirrored_walk_list, 14)
205 tenor_5_random_walk_notes = [x for x in randomWalk(
206     random_seed=15,
207     length=1000,
208     step_list=[1],
209     mapped_list=tenor_5_walk_chord
210     )
211   ]
212
213 baritone_1_walk_chord = rotate(mirrored_walk_list, 28)
214 baritone_1_random_walk_notes = [x for x in randomWalk(
```

```

215     random_seed=16,
216     length=1000,
217     step_list=[1],
218     mapped_list=baritone_1_walk_chord
219     )
220   ]
221
222 baritone_2_walk_chord = rotate(mirrored_walk_list, 14)
223 baritone_2_random_walk_notes = [x for x in randomWalk(
224   random_seed=17,
225   length=1000,
226   step_list=[1],
227   mapped_list=baritone_2_walk_chord
228   )
229   ]
230
231 baritone_3_walk_chord = rotate(mirrored_walk_list, 10)
232 baritone_3_random_walk_notes = [x for x in randomWalk(
233   random_seed=18,
234   length=1000,
235   step_list=[1],
236   mapped_list=baritone_3_walk_chord
237   )
238   ]
239
240 bass_1_walk_chord = rotate(mirrored_walk_list, 24)
241 bass_1_random_walk_notes = [x for x in randomWalk(
242   random_seed=19,
243   length=1000,
244   step_list=[1],
245   mapped_list=bass_1_walk_chord
246   )
247   ]
248
249 bass_2_walk_chord = rotate(mirrored_walk_list, 20)
250 bass_2_random_walk_notes = [x for x in randomWalk(
251   random_seed=20,
252   length=1000,
253   step_list=[1],
254   mapped_list=bass_2_walk_chord
255   )
256   ]
257
258 contrabass_walk_chord = rotate(mirrored_walk_list, 6)
259 contrabass_random_walk_notes = [x for x in randomWalk(
260   random_seed=21,
261   length=1000,
262   step_list=[1],
263   mapped_list=contrabass_walk_chord
264   )
265   ]
266
267 rmaker_one = abjadext.rmakers.NoteRhythmMaker()
268
269 rmaker_two = abjadext.rmakers.TaleaRhythmMaker(
270   talea=abjadext.rmakers.Talea(
271     counts=[1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 3, 2, 2, 3, 2, 1, 1, 2, 1, 1, 3,
272   ],
273     denominator=16,
```

```

273     ),
274     beamSpecifier=abjadext.rmakers.BeamSpecifier(
275         beamDivisionsTogether=True,
276         beamRests=False,
277     ),
278     extraCountsPerDivision=[-1, 0, 1, -1, 1, 0, ],
279     logicalTieMasks=[
280         abjadext.rmakers.silence([8], 5),
281     ],
282     divisionMasks=[
283         abjadext.rmakers.SilenceMask(
284             pattern=abjad.index([2], 11),
285         ),
286     ],
287     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
288         trivialize=True,
289         extractTrivial=True,
290         rewriteRestFilled=True,
291         rewriteSustained=True,
292     ),
293 )
294
295 attachmentHandlerOne = AttachmentHandler(
296     startingDynamic='mp',
297     endingDynamic='f',
298     hairpin='<',
299 )
300
301 attachmentHandlerTwo = AttachmentHandler(
302     startingDynamic='mf',
303     endingDynamic='pp',
304     hairpin='>',
305 )
306
307 #####sopranino#####
308 sopraninoMusicmakerOne = MusicMaker(
309     rmaker=rmakerOne,
310     pitches=sopraninoChord,
311     continuous=True,
312     attachmentHandler=attachmentHandlerOne,
313 )
314 sopraninoMusicmakerTwo = MusicMaker(
315     rmaker=rmakerTwo,
316     pitches=sopraninoRandomWalkNotes,
317     continuous=True,
318     attachmentHandler=attachmentHandlerOne,
319 )
320 sopraninoMusicmakerThree = MusicMaker(
321     rmaker=rmakerTwo,
322     pitches=sopraninoRandomWalkNotes,
323     continuous=True,
324     attachmentHandler=attachmentHandlerTwo,
325 )
326 #####soprano_one#####
327 sopranoOneMusicmakerOne = MusicMaker(
328     rmaker=rmakerOne,
329     pitches=soprano1Chord,
330     continuous=True,
331     attachmentHandler=attachmentHandlerOne,

```

```
332 )
333 soprano_one_musicmaker_two = MusicMaker(
334     rmaker=rmaker_two,
335     pitches=soprano_1_random_walk_notes,
336     continuous=True,
337     attachment_handler=attachment_handler_one,
338 )
339 soprano_one_musicmaker_three = MusicMaker(
340     rmaker=rmaker_two,
341     pitches=soprano_1_random_walk_notes,
342     continuous=True,
343     attachment_handler=attachment_handler_two,
344 )
345 #####soprano_two#####
346 soprano_two_musicmaker_one = MusicMaker(
347     rmaker=rmaker_one,
348     pitches=soprano_2_chord,
349     continuous=True,
350     attachment_handler=attachment_handler_one,
351 )
352 soprano_two_musicmaker_two = MusicMaker(
353     rmaker=rmaker_two,
354     pitches=soprano_2_random_walk_notes,
355     continuous=True,
356     attachment_handler=attachment_handler_one,
357 )
358 soprano_two_musicmaker_three = MusicMaker(
359     rmaker=rmaker_two,
360     pitches=soprano_2_random_walk_notes,
361     continuous=True,
362     attachment_handler=attachment_handler_two,
363 )
364 #####soprano_three#####
365 soprano_three_musicmaker_one = MusicMaker(
366     rmaker=rmaker_one,
367     pitches=soprano_3_chord,
368     continuous=True,
369     attachment_handler=attachment_handler_one,
370 )
371 soprano_three_musicmaker_two = MusicMaker(
372     rmaker=rmaker_two,
373     pitches=soprano_3_random_walk_notes,
374     continuous=True,
375     attachment_handler=attachment_handler_one,
376 )
377 soprano_three_musicmaker_three = MusicMaker(
378     rmaker=rmaker_two,
379     pitches=soprano_3_random_walk_notes,
380     continuous=True,
381     attachment_handler=attachment_handler_two,
382 )
383 #####alto_one#####
384 alto_one_musicmaker_one = MusicMaker(
385     rmaker=rmaker_one,
386     pitches=alto_1_chord,
387     continuous=True,
388     attachment_handler=attachment_handler_one,
389 )
390 alto_one_musicmaker_two = MusicMaker(
```

```
391     rmaker=rmaker_two,
392     pitches=soprano_1_random_walk_notes,
393     continuous=True,
394     attachment_handler=attachment_handler_one,
395 )
396 alto_one_musicmaker_three = MusicMaker(
397     rmaker=rmaker_two,
398     pitches=soprano_1_random_walk_notes,
399     continuous=True,
400     attachment_handler=attachment_handler_two,
401 )
402 #####alto_two#####
403 alto_two_musicmaker_one = MusicMaker(
404     rmaker=rmaker_one,
405     pitches=alto_2_chord,
406     continuous=True,
407     attachment_handler=attachment_handler_one,
408 )
409 alto_two_musicmaker_two = MusicMaker(
410     rmaker=rmaker_two,
411     pitches=soprano_2_random_walk_notes,
412     continuous=True,
413     attachment_handler=attachment_handler_one,
414 )
415 alto_two_musicmaker_three = MusicMaker(
416     rmaker=rmaker_two,
417     pitches=soprano_2_random_walk_notes,
418     continuous=True,
419     attachment_handler=attachment_handler_two,
420 )
421 #####alto_three#####
422 alto_three_musicmaker_one = MusicMaker(
423     rmaker=rmaker_one,
424     pitches=alto_3_chord,
425     continuous=True,
426     attachment_handler=attachment_handler_one,
427 )
428 alto_three_musicmaker_two = MusicMaker(
429     rmaker=rmaker_two,
430     pitches=soprano_3_random_walk_notes,
431     continuous=True,
432     attachment_handler=attachment_handler_one,
433 )
434 alto_three_musicmaker_three = MusicMaker(
435     rmaker=rmaker_two,
436     pitches=soprano_3_random_walk_notes,
437     continuous=True,
438     attachment_handler=attachment_handler_two,
439 )
440 #####alto_four#####
441 alto_four_musicmaker_one = MusicMaker(
442     rmaker=rmaker_one,
443     pitches=alto_4_chord,
444     continuous=True,
445     attachment_handler=attachment_handler_one,
446 )
447 alto_four_musicmaker_two = MusicMaker(
448     rmaker=rmaker_two,
449     pitches=alto_4_random_walk_notes,
```

```
450     continuous=True,
451     attachment_handler=attachment_handler_one,
452 )
453 alto_four_musicmaker_three = MusicMaker(
454     rmaker=rmaker_two,
455     pitches=alto_4_random_walk_notes,
456     continuous=True,
457     attachment_handler=attachment_handler_two,
458 )
459 #####alto_five#####
460 alto_five_musicmaker_one = MusicMaker(
461     rmaker=rmaker_one,
462     pitches=alto_5_chord,
463     continuous=True,
464     attachment_handler=attachment_handler_one,
465 )
466 alto_five_musicmaker_two = MusicMaker(
467     rmaker=rmaker_two,
468     pitches=alto_5_random_walk_notes,
469     continuous=True,
470     attachment_handler=attachment_handler_one,
471 )
472 alto_five_musicmaker_three = MusicMaker(
473     rmaker=rmaker_two,
474     pitches=alto_5_random_walk_notes,
475     continuous=True,
476     attachment_handler=attachment_handler_two,
477 )
478 #####alto_six#####
479 alto_six_musicmaker_one = MusicMaker(
480     rmaker=rmaker_one,
481     pitches=alto_6_chord,
482     continuous=True,
483     attachment_handler=attachment_handler_one,
484 )
485 alto_six_musicmaker_two = MusicMaker(
486     rmaker=rmaker_two,
487     pitches=alto_6_random_walk_notes,
488     continuous=True,
489     attachment_handler=attachment_handler_one,
490 )
491 alto_six_musicmaker_three = MusicMaker(
492     rmaker=rmaker_two,
493     pitches=alto_6_random_walk_notes,
494     continuous=True,
495     attachment_handler=attachment_handler_two,
496 )
497 #####tenor_one#####
498 tenor_one_musicmaker_one = MusicMaker(
499     rmaker=rmaker_one,
500     pitches=tenor_1_chord,
501     continuous=True,
502     attachment_handler=attachment_handler_one,
503 )
504 tenor_one_musicmaker_two = MusicMaker(
505     rmaker=rmaker_two,
506     pitches=tenor_1_random_walk_notes,
507     continuous=True,
508     attachment_handler=attachment_handler_one,
```

```
509 )
510 tenor_one_musicmaker_three = MusicMaker(
511     rmaker=rmaker_two,
512     pitches=tenor_1_random_walk_notes,
513     continuous=True,
514     attachment_handler=attachment_handler_two,
515 )
516 #####tenor_two#####
517 tenor_two_musicmaker_one = MusicMaker(
518     rmaker=rmaker_one,
519     pitches=tenor_2_chord,
520     continuous=True,
521     attachment_handler=attachment_handler_one,
522 )
523 tenor_two_musicmaker_two = MusicMaker(
524     rmaker=rmaker_two,
525     pitches=tenor_2_random_walk_notes,
526     continuous=True,
527     attachment_handler=attachment_handler_one,
528 )
529 tenor_two_musicmaker_three = MusicMaker(
530     rmaker=rmaker_two,
531     pitches=tenor_2_random_walk_notes,
532     continuous=True,
533     attachment_handler=attachment_handler_two,
534 )
535 #####tenor_three#####
536 tenor_three_musicmaker_one = MusicMaker(
537     rmaker=rmaker_one,
538     pitches=tenor_3_chord,
539     continuous=True,
540     attachment_handler=attachment_handler_one,
541 )
542 tenor_three_musicmaker_two = MusicMaker(
543     rmaker=rmaker_two,
544     pitches=tenor_3_random_walk_notes,
545     continuous=True,
546     attachment_handler=attachment_handler_one,
547 )
548 tenor_three_musicmaker_three = MusicMaker(
549     rmaker=rmaker_two,
550     pitches=tenor_3_random_walk_notes,
551     continuous=True,
552     attachment_handler=attachment_handler_two,
553 )
554 #####tenor_four#####
555 tenor_four_musicmaker_one = MusicMaker(
556     rmaker=rmaker_one,
557     pitches=tenor_4_chord,
558     continuous=True,
559     attachment_handler=attachment_handler_one,
560 )
561 tenor_four_musicmaker_two = MusicMaker(
562     rmaker=rmaker_two,
563     pitches=tenor_4_random_walk_notes,
564     continuous=True,
565     attachment_handler=attachment_handler_one,
566 )
567 tenor_four_musicmaker_three = MusicMaker(
```

```
568     rmaker=rmaker_two,
569     pitches=tenor_4_random_walk_notes,
570     continuous=True,
571     attachment_handler=attachment_handler_two,
572 )
573 #####tenor#####
574 tenor_five_musicmaker_one = MusicMaker(
575     rmaker=rmaker_one,
576     pitches=tenor_5_chord,
577     continuous=True,
578     attachment_handler=attachment_handler_one,
579 )
580 tenor_five_musicmaker_two = MusicMaker(
581     rmaker=rmaker_two,
582     pitches=tenor_5_random_walk_notes,
583     continuous=True,
584     attachment_handler=attachment_handler_one,
585 )
586 tenor_five_musicmaker_three = MusicMaker(
587     rmaker=rmaker_two,
588     pitches=tenor_5_random_walk_notes,
589     continuous=True,
590     attachment_handler=attachment_handler_two,
591 )
592 #####baritone#####
593 baritone_one_musicmaker_one = MusicMaker(
594     rmaker=rmaker_one,
595     pitches=baritone_1_chord,
596     continuous=True,
597     attachment_handler=attachment_handler_one,
598 )
599 baritone_one_musicmaker_two = MusicMaker(
600     rmaker=rmaker_two,
601     pitches=baritone_1_random_walk_notes,
602     continuous=True,
603     attachment_handler=attachment_handler_one,
604 )
605 baritone_one_musicmaker_three = MusicMaker(
606     rmaker=rmaker_two,
607     pitches=baritone_1_random_walk_notes,
608     continuous=True,
609     attachment_handler=attachment_handler_two,
610 )
611 #####baritone#####
612 baritone_two_musicmaker_one = MusicMaker(
613     rmaker=rmaker_one,
614     pitches=baritone_2_chord,
615     continuous=True,
616     attachment_handler=attachment_handler_one,
617 )
618 baritone_two_musicmaker_two = MusicMaker(
619     rmaker=rmaker_two,
620     pitches=baritone_2_random_walk_notes,
621     continuous=True,
622     attachment_handler=attachment_handler_one,
623 )
624 baritone_two_musicmaker_three = MusicMaker(
625     rmaker=rmaker_two,
626     pitches=baritone_2_random_walk_notes,
```

```
627     continuous=True,
628     attachment_handler=attachment_handler_two,
629 )
630 #####baritone_three#####
631 baritone_three_musicmaker_one = MusicMaker(
632     rmaker=rmaker_one,
633     pitches=baritone_3_chord,
634     continuous=True,
635     attachment_handler=attachment_handler_one,
636 )
637 baritone_three_musicmaker_two = MusicMaker(
638     rmaker=rmaker_two,
639     pitches=baritone_3_random_walk_notes,
640     continuous=True,
641     attachment_handler=attachment_handler_one,
642 )
643 baritone_three_musicmaker_three = MusicMaker(
644     rmaker=rmaker_two,
645     pitches=baritone_3_random_walk_notes,
646     continuous=True,
647     attachment_handler=attachment_handler_two,
648 )
649 #####bass_one#####
650 bass_one_musicmaker_one = MusicMaker(
651     rmaker=rmaker_one,
652     pitches=bass_1_chord,
653     continuous=True,
654     attachment_handler=attachment_handler_one,
655 )
656 bass_one_musicmaker_two = MusicMaker(
657     rmaker=rmaker_two,
658     pitches=bass_1_random_walk_notes,
659     continuous=True,
660     attachment_handler=attachment_handler_one,
661 )
662 bass_one_musicmaker_three = MusicMaker(
663     rmaker=rmaker_two,
664     pitches=bass_1_random_walk_notes,
665     continuous=True,
666     attachment_handler=attachment_handler_two,
667 )
668 #####bass_two#####
669 bass_two_musicmaker_one = MusicMaker(
670     rmaker=rmaker_one,
671     pitches=bass_2_chord,
672     continuous=True,
673     attachment_handler=attachment_handler_one,
674 )
675 bass_two_musicmaker_two = MusicMaker(
676     rmaker=rmaker_two,
677     pitches=bass_2_random_walk_notes,
678     continuous=True,
679     attachment_handler=attachment_handler_one,
680 )
681 bass_two_musicmaker_three = MusicMaker(
682     rmaker=rmaker_two,
683     pitches=bass_2_random_walk_notes,
684     continuous=True,
685     attachment_handler=attachment_handler_two,
```

```

686 )
687 #####contrabass#####
688 contrabass_musicmaker_one = MusicMaker(
689     rmaker=rmaker_one,
690     pitches=contrabass_chord,
691     continuous=True,
692     attachment_handler=attachment_handler_one,
693 )
694 contrabass_musicmaker_two = MusicMaker(
695     rmaker=rmaker_two,
696     pitches=contrabass_random_walk_notes,
697     continuous=True,
698     attachment_handler=attachment_handler_one,
699 )
700 contrabass_musicmaker_three = MusicMaker(
701     rmaker=rmaker_two,
702     pitches=contrabass_random_walk_notes,
703     continuous=True,
704     attachment_handler=attachment_handler_two,
705 )
706
707 silence_maker = abjadext.rmakers.NoteRhythmMaker(
708     division_masks=[
709         abjadext.rmakers.SilenceMask(
710             pattern=abjad.index([0], 1),
711             ),
712         ],
713     )
714
715 class MusicSpecifier:
716
717     def __init__(self, music_maker, voice_name):
718         self.music_maker = music_maker
719         self.voice_name = voice_name
720
721     print('Collecting timespans and rmakers ...')
722
723     voice_1_timespan_list = abjad.TimespanList([
724         abjad.AnnotatedTimespan(
725             start_offset=start_offset,
726             stop_offset=stop_offset,
727             annotation=MusicSpecifier(
728                 music_maker=music_maker,
729                 voice_name='Voice 1',
730             ),
731         )
732         for start_offset, stop_offset, music_maker in [
733             [(0, 8), (2, 8), soprano_musicmaker_two],
734             [(2, 8), (4, 8), soprano_musicmaker_three],
735             [(4, 8), (6, 8), soprano_musicmaker_two],
736             [(6, 8), (8, 8), soprano_musicmaker_two],
737             [(8, 8), (10, 8), soprano_musicmaker_three],
738             [(10, 8), (12, 8), soprano_musicmaker_two],
739             [(12, 8), (14, 8), soprano_musicmaker_two],
740             [(14, 8), (16, 8), soprano_musicmaker_two],
741             [(16, 8), (18, 8), soprano_musicmaker_three],
742             [(18, 8), (20, 8), soprano_musicmaker_two],
743         ]
744     )

```

```

745     [(20, 8), (22, 8), sopranino_musicmaker_two],
746     [(22, 8), (24, 8), sopranino_musicmaker_two],
747
748     [(24, 8), (26, 8), sopranino_musicmaker_two],
749     [(26, 8), (28, 8), sopranino_musicmaker_three],
750     [(28, 8), (30, 8), sopranino_musicmaker_two],
751     [(30, 8), (32, 8), sopranino_musicmaker_two],
752
753     [(32, 8), (34, 8), sopranino_musicmaker_two],
754     [(34, 8), (36, 8), sopranino_musicmaker_two],
755     [(36, 8), (38, 8), sopranino_musicmaker_three],
756     [(38, 8), (40, 8), sopranino_musicmaker_three],
757
758     [(40, 8), (42, 8), sopranino_musicmaker_two],
759     [(42, 8), (44, 8), sopranino_musicmaker_two],
760     [(44, 8), (46, 8), sopranino_musicmaker_two],
761     [(46, 8), (48, 8), sopranino_musicmaker_three],
762
763     [(48, 8), (50, 8), sopranino_musicmaker_three],
764     [(50, 8), (52, 8), sopranino_musicmaker_three],
765     [(52, 8), (54, 8), sopranino_musicmaker_two],
766     [(54, 8), (56, 8), sopranino_musicmaker_two],
767
768     [(56, 8), (58, 8), sopranino_musicmaker_three],
769     [(58, 8), (60, 8), sopranino_musicmaker_three],
770     [(60, 8), (62, 8), sopranino_musicmaker_three],
771     [(62, 8), (64, 8), sopranino_musicmaker_two],
772
773     [(64, 8), (66, 8), sopranino_musicmaker_two],
774     [(66, 8), (68, 8), sopranino_musicmaker_three],
775     [(68, 8), (70, 8), sopranino_musicmaker_three],
776     [(70, 8), (72, 8), sopranino_musicmaker_two],
777
778     [(72, 8), (74, 8), sopranino_musicmaker_three],
779     [(74, 8), (76, 8), sopranino_musicmaker_three],
780     [(76, 8), (78, 8), sopranino_musicmaker_two],
781     [(78, 8), (80, 8), sopranino_musicmaker_three],
782
783     [(80, 8), (82, 8), sopranino_musicmaker_three],
784     [(82, 8), (84, 8), sopranino_musicmaker_three],
785     [(84, 8), (86, 8), sopranino_musicmaker_three],
786     [(86, 8), (88, 8), sopranino_musicmaker_two],
787
788     [(88, 8), (90, 8), sopranino_musicmaker_two],
789     [(90, 8), (92, 8), sopranino_musicmaker_two],
790     [(92, 8), (94, 8), sopranino_musicmaker_two],
791     [(94, 8), (96, 8), sopranino_musicmaker_three],
792
793     [(96, 8), (98, 8), sopranino_musicmaker_three],
794     [(98, 8), (100, 8), sopranino_musicmaker_three],
795     [(100, 8), (102, 8), sopranino_musicmaker_three],
796     [(102, 8), (104, 8), sopranino_musicmaker_two],
797 ]
798 ])
799
800 voice_2_timespan_list = abjad.TimespanList([
801     abjad.AnnotatedTimespan(
802         start_offset=start_offset,
803         stop_offset=stop_offset,

```

```

804     annotation=MusicSpecifier(
805         music_maker=music_maker,
806         voice_name='Voice 2',
807     ),
808 )
809 for start_offset, stop_offset, music_maker in [
810     [(0, 8), (2, 8), soprano_one_musicmaker_two],
811     [(2, 8), (4, 8), soprano_one_musicmaker_two],
812     [(4, 8), (6, 8), soprano_one_musicmaker_two],
813     [(6, 8), (8, 8), soprano_one_musicmaker_two],
814
815     [(8, 8), (10, 8), soprano_one_musicmaker_two],
816     [(10, 8), (12, 8), soprano_one_musicmaker_three],
817     [(12, 8), (14, 8), soprano_one_musicmaker_two],
818     [(14, 8), (16, 8), soprano_one_musicmaker_three],
819
820     [(16, 8), (18, 8), soprano_one_musicmaker_three],
821     [(18, 8), (20, 8), soprano_one_musicmaker_three],
822     [(20, 8), (22, 8), soprano_one_musicmaker_two],
823     [(22, 8), (24, 8), soprano_one_musicmaker_three],
824
825     [(24, 8), (26, 8), soprano_one_musicmaker_three],
826     [(26, 8), (28, 8), soprano_one_musicmaker_three],
827     [(28, 8), (30, 8), soprano_one_musicmaker_two],
828     [(30, 8), (32, 8), soprano_one_musicmaker_two],
829
830     [(32, 8), (34, 8), soprano_one_musicmaker_two],
831     [(34, 8), (36, 8), soprano_one_musicmaker_two],
832     [(36, 8), (38, 8), soprano_one_musicmaker_two],
833     [(38, 8), (40, 8), soprano_one_musicmaker_three],
834
835     [(40, 8), (42, 8), soprano_one_musicmaker_three],
836     [(42, 8), (44, 8), soprano_one_musicmaker_three],
837     [(44, 8), (46, 8), soprano_one_musicmaker_three],
838     [(46, 8), (48, 8), soprano_one_musicmaker_two],
839
840     [(48, 8), (50, 8), soprano_one_musicmaker_two],
841     [(50, 8), (52, 8), soprano_one_musicmaker_two],
842     [(52, 8), (54, 8), soprano_one_musicmaker_two],
843     [(54, 8), (56, 8), soprano_one_musicmaker_three],
844
845     [(56, 8), (58, 8), soprano_one_musicmaker_three],
846     [(58, 8), (60, 8), soprano_one_musicmaker_three],
847     [(60, 8), (62, 8), soprano_one_musicmaker_two],
848     [(62, 8), (64, 8), soprano_one_musicmaker_three],
849
850     [(64, 8), (66, 8), soprano_one_musicmaker_three],
851     [(66, 8), (68, 8), soprano_one_musicmaker_three],
852     [(68, 8), (70, 8), soprano_one_musicmaker_two],
853     [(70, 8), (72, 8), soprano_one_musicmaker_two],
854
855     [(72, 8), (74, 8), soprano_one_musicmaker_two],
856     [(74, 8), (76, 8), soprano_one_musicmaker_two],
857     [(76, 8), (78, 8), soprano_one_musicmaker_three],
858     [(78, 8), (80, 8), soprano_one_musicmaker_three],
859
860     [(80, 8), (82, 8), soprano_one_musicmaker_three],
861     [(82, 8), (84, 8), soprano_one_musicmaker_two],
862     [(84, 8), (86, 8), soprano_one_musicmaker_two],

```

```

863     [(86, 8), (88, 8), soprano_one_musicmaker_three],
864
865     [(88, 8), (90, 8), soprano_one_musicmaker_two],
866     [(90, 8), (92, 8), soprano_one_musicmaker_two],
867     [(92, 8), (94, 8), soprano_one_musicmaker_two],
868     [(94, 8), (96, 8), soprano_one_musicmaker_three],
869
870     [(96, 8), (98, 8), soprano_one_musicmaker_three],
871     [(98, 8), (100, 8), soprano_one_musicmaker_three],
872     [(100, 8), (102, 8), soprano_one_musicmaker_two],
873     [(102, 8), (104, 8), soprano_one_musicmaker_two],
874 ]
875 ])
876
877 voice_3_timespan_list = abjad.TimespanList([
878     abjad.AnnotatedTimespan(
879         start_offset=start_offset,
880         stop_offset=stop_offset,
881         annotation=MusicSpecifier(
882             music_maker=music_maker,
883             voice_name='Voice 3',
884         ),
885     )
886     for start_offset, stop_offset, music_maker in [
887         [(0, 8), (2, 8), soprano_two_musicmaker_three],
888         [(2, 8), (4, 8), soprano_two_musicmaker_three],
889         [(4, 8), (6, 8), soprano_two_musicmaker_two],
890         [(6, 8), (8, 8), soprano_two_musicmaker_two],
891
892         [(8, 8), (10, 8), soprano_two_musicmaker_three],
893         [(10, 8), (12, 8), soprano_two_musicmaker_three],
894         [(12, 8), (14, 8), soprano_two_musicmaker_two],
895         [(14, 8), (16, 8), soprano_two_musicmaker_three],
896
897         [(16, 8), (18, 8), soprano_two_musicmaker_two],
898         [(18, 8), (20, 8), soprano_two_musicmaker_three],
899         [(20, 8), (22, 8), soprano_two_musicmaker_two],
900         [(22, 8), (24, 8), soprano_two_musicmaker_two],
901
902         [(24, 8), (26, 8), soprano_two_musicmaker_three],
903         [(26, 8), (28, 8), soprano_two_musicmaker_three],
904         [(28, 8), (30, 8), soprano_two_musicmaker_two],
905         [(30, 8), (32, 8), soprano_two_musicmaker_three],
906
907         [(32, 8), (34, 8), soprano_two_musicmaker_three],
908         [(34, 8), (36, 8), soprano_two_musicmaker_three],
909         [(36, 8), (38, 8), soprano_two_musicmaker_three],
910         [(38, 8), (40, 8), soprano_two_musicmaker_two],
911
912         [(40, 8), (42, 8), soprano_two_musicmaker_three],
913         [(42, 8), (44, 8), soprano_two_musicmaker_two],
914         [(44, 8), (46, 8), soprano_two_musicmaker_two],
915         [(46, 8), (48, 8), soprano_two_musicmaker_three],
916
917         [(48, 8), (50, 8), soprano_two_musicmaker_two],
918         [(50, 8), (52, 8), soprano_two_musicmaker_three],
919         [(52, 8), (54, 8), soprano_two_musicmaker_three],
920         [(54, 8), (56, 8), soprano_two_musicmaker_two],
921

```

```

922     [(56, 8), (58, 8), soprano_two_musicmaker_three],
923     [(58, 8), (60, 8), soprano_two_musicmaker_two],
924     [(60, 8), (62, 8), soprano_two_musicmaker_three],
925     [(62, 8), (64, 8), soprano_two_musicmaker_two],
926
927     [(64, 8), (66, 8), soprano_two_musicmaker_three],
928     [(66, 8), (68, 8), soprano_two_musicmaker_three],
929     [(68, 8), (70, 8), soprano_two_musicmaker_two],
930     [(70, 8), (72, 8), soprano_two_musicmaker_three],
931
932     [(72, 8), (74, 8), soprano_two_musicmaker_two],
933     [(74, 8), (76, 8), soprano_two_musicmaker_two],
934     [(76, 8), (78, 8), soprano_two_musicmaker_three],
935     [(78, 8), (80, 8), soprano_two_musicmaker_three],
936
937     [(80, 8), (82, 8), soprano_two_musicmaker_three],
938     [(82, 8), (84, 8), soprano_two_musicmaker_two],
939     [(84, 8), (86, 8), soprano_two_musicmaker_two],
940     [(86, 8), (88, 8), soprano_two_musicmaker_two],
941
942     [(88, 8), (90, 8), soprano_two_musicmaker_two],
943     [(90, 8), (92, 8), soprano_two_musicmaker_three],
944     [(92, 8), (94, 8), soprano_two_musicmaker_three],
945     [(94, 8), (96, 8), soprano_two_musicmaker_three],
946
947     [(96, 8), (98, 8), soprano_two_musicmaker_three],
948     [(98, 8), (100, 8), soprano_two_musicmaker_two],
949     [(100, 8), (102, 8), soprano_two_musicmaker_two],
950     [(102, 8), (104, 8), soprano_two_musicmaker_two],
951 ]
952 ])
953
954 voice_4_timespan_list = abjad.TimespanList([
955     abjad.AnnotatedTimespan(
956         start_offset=start_offset,
957         stop_offset=stop_offset,
958         annotation=MusicSpecifier(
959             music_maker=music_maker,
960             voice_name='Voice 4',
961         ),
962     )
963     for start_offset, stop_offset, music_maker in [
964         [(0, 8), (2, 8), soprano_three_musicmaker_three],
965         [(2, 8), (4, 8), soprano_three_musicmaker_two],
966         [(4, 8), (6, 8), soprano_three_musicmaker_two],
967         [(6, 8), (8, 8), soprano_three_musicmaker_two],
968
969         [(8, 8), (10, 8), soprano_three_musicmaker_three],
970         [(10, 8), (12, 8), soprano_three_musicmaker_three],
971         [(12, 8), (14, 8), soprano_three_musicmaker_three],
972         [(14, 8), (16, 8), soprano_three_musicmaker_three],
973
974         [(16, 8), (18, 8), soprano_three_musicmaker_two],
975         [(18, 8), (20, 8), soprano_three_musicmaker_two],
976         [(20, 8), (22, 8), soprano_three_musicmaker_two],
977         [(22, 8), (24, 8), soprano_three_musicmaker_two],
978
979         [(24, 8), (26, 8), soprano_three_musicmaker_two],
980         [(26, 8), (28, 8), soprano_three_musicmaker_three],

```

```

981     [(28, 8), (30, 8), soprano_three_musicmaker_two],
982     [(30, 8), (32, 8), soprano_three_musicmaker_three],
983
984     [(32, 8), (34, 8), soprano_three_musicmaker_three],
985     [(34, 8), (36, 8), soprano_three_musicmaker_two],
986     [(36, 8), (38, 8), soprano_three_musicmaker_two],
987     [(38, 8), (40, 8), soprano_three_musicmaker_three],
988
989     [(40, 8), (42, 8), soprano_three_musicmaker_three],
990     [(42, 8), (44, 8), soprano_three_musicmaker_three],
991     [(44, 8), (46, 8), soprano_three_musicmaker_two],
992     [(46, 8), (48, 8), soprano_three_musicmaker_two],
993
994     [(48, 8), (50, 8), soprano_three_musicmaker_two],
995     [(50, 8), (52, 8), soprano_three_musicmaker_two],
996     [(52, 8), (54, 8), soprano_three_musicmaker_two],
997     [(54, 8), (56, 8), soprano_three_musicmaker_three],
998
999     [(56, 8), (58, 8), soprano_three_musicmaker_two],
1000    [(58, 8), (60, 8), soprano_three_musicmaker_three],
1001    [(60, 8), (62, 8), soprano_three_musicmaker_three],
1002    [(62, 8), (64, 8), soprano_three_musicmaker_three],
1003
1004    [(64, 8), (66, 8), soprano_three_musicmaker_three],
1005    [(66, 8), (68, 8), soprano_three_musicmaker_three],
1006    [(68, 8), (70, 8), soprano_three_musicmaker_two],
1007    [(70, 8), (72, 8), soprano_three_musicmaker_three],
1008
1009    [(72, 8), (74, 8), soprano_three_musicmaker_two],
1010    [(74, 8), (76, 8), soprano_three_musicmaker_three],
1011    [(76, 8), (78, 8), soprano_three_musicmaker_two],
1012    [(78, 8), (80, 8), soprano_three_musicmaker_two],
1013
1014    [(80, 8), (82, 8), soprano_three_musicmaker_two],
1015    [(82, 8), (84, 8), soprano_three_musicmaker_two],
1016    [(84, 8), (86, 8), soprano_three_musicmaker_three],
1017    [(86, 8), (88, 8), soprano_three_musicmaker_three],
1018
1019    [(88, 8), (90, 8), soprano_three_musicmaker_three],
1020    [(90, 8), (92, 8), soprano_three_musicmaker_three],
1021    [(92, 8), (94, 8), soprano_three_musicmaker_three],
1022    [(94, 8), (96, 8), soprano_three_musicmaker_two],
1023
1024    [(96, 8), (98, 8), soprano_three_musicmaker_three],
1025    [(98, 8), (100, 8), soprano_three_musicmaker_three],
1026    [(100, 8), (102, 8), soprano_three_musicmaker_two],
1027    [(102, 8), (104, 8), soprano_three_musicmaker_three],
1028 ]
1029 ])
1030
1031 voice_5_timespan_list = abjad.TimespanList([
1032     abjad.AnnotatedTimespan(
1033         start_offset=start_offset,
1034         stop_offset=stop_offset,
1035         annotation=MusicSpecifier(
1036             music_maker=music_maker,
1037             voice_name='Voice 5',
1038         ),
1039     )

```

```
1040 for start_offset, stop_offset, music_maker in [
1041     [(0, 8), (2, 8), alto_one_musicmaker_three],
1042     [(2, 8), (4, 8), alto_one_musicmaker_three],
1043     [(4, 8), (6, 8), alto_one_musicmaker_two],
1044     [(6, 8), (8, 8), alto_one_musicmaker_two],
1045
1046     [(8, 8), (10, 8), alto_one_musicmaker_two],
1047     [(10, 8), (12, 8), alto_one_musicmaker_three],
1048     [(12, 8), (14, 8), alto_one_musicmaker_two],
1049     [(14, 8), (16, 8), alto_one_musicmaker_three],
1050
1051     [(16, 8), (18, 8), alto_one_musicmaker_three],
1052     [(18, 8), (20, 8), alto_one_musicmaker_three],
1053     [(20, 8), (22, 8), alto_one_musicmaker_three],
1054     [(22, 8), (24, 8), alto_one_musicmaker_three],
1055
1056     [(24, 8), (26, 8), alto_one_musicmaker_two],
1057     [(26, 8), (28, 8), alto_one_musicmaker_two],
1058     [(28, 8), (30, 8), alto_one_musicmaker_two],
1059     [(30, 8), (32, 8), alto_one_musicmaker_two],
1060
1061     [(32, 8), (34, 8), alto_one_musicmaker_two],
1062     [(34, 8), (36, 8), alto_one_musicmaker_three],
1063     [(36, 8), (38, 8), alto_one_musicmaker_three],
1064     [(38, 8), (40, 8), alto_one_musicmaker_three],
1065
1066     [(40, 8), (42, 8), alto_one_musicmaker_two],
1067     [(42, 8), (44, 8), alto_one_musicmaker_three],
1068     [(44, 8), (46, 8), alto_one_musicmaker_two],
1069     [(46, 8), (48, 8), alto_one_musicmaker_two],
1070
1071     [(48, 8), (50, 8), alto_one_musicmaker_two],
1072     [(50, 8), (52, 8), alto_one_musicmaker_three],
1073     [(52, 8), (54, 8), alto_one_musicmaker_two],
1074     [(54, 8), (56, 8), alto_one_musicmaker_three],
1075
1076     [(56, 8), (58, 8), alto_one_musicmaker_three],
1077     [(58, 8), (60, 8), alto_one_musicmaker_three],
1078     [(60, 8), (62, 8), alto_one_musicmaker_two],
1079     [(62, 8), (64, 8), alto_one_musicmaker_three],
1080
1081     [(64, 8), (66, 8), alto_one_musicmaker_two],
1082     [(66, 8), (68, 8), alto_one_musicmaker_three],
1083     [(68, 8), (70, 8), alto_one_musicmaker_three],
1084     [(70, 8), (72, 8), alto_one_musicmaker_two],
1085
1086     [(72, 8), (74, 8), alto_one_musicmaker_two],
1087     [(74, 8), (76, 8), alto_one_musicmaker_three],
1088     [(76, 8), (78, 8), alto_one_musicmaker_three],
1089     [(78, 8), (80, 8), alto_one_musicmaker_three],
1090
1091     [(80, 8), (82, 8), alto_one_musicmaker_three],
1092     [(82, 8), (84, 8), alto_one_musicmaker_two],
1093     [(84, 8), (86, 8), alto_one_musicmaker_two],
1094     [(86, 8), (88, 8), alto_one_musicmaker_two],
1095
1096     [(88, 8), (90, 8), alto_one_musicmaker_two],
1097     [(90, 8), (92, 8), alto_one_musicmaker_three],
1098     [(92, 8), (94, 8), alto_one_musicmaker_two],
```

```

1099     [(94, 8), (96, 8), alto_one_musicmaker_three],
1100
1101     [(96, 8), (98, 8), alto_one_musicmaker_three],
1102     [(98, 8), (100, 8), alto_one_musicmaker_two],
1103     [(100, 8), (102, 8), alto_one_musicmaker_two],
1104     [(102, 8), (104, 8), alto_one_musicmaker_three],
1105 ]
1106 ])
1107
1108 voice_6_timestspan_list = abjad.TimespanList([
1109     abjad.AnnotatedTimespan(
1110         start_offset=start_offset,
1111         stop_offset=stop_offset,
1112         annotation=MusicSpecifier(
1113             music_maker=music_maker,
1114             voice_name='Voice 6',
1115         ),
1116     )
1117     for start_offset, stop_offset, music_maker in [
1118         [(0, 8), (2, 8), alto_two_musicmaker_three],
1119         [(2, 8), (4, 8), alto_two_musicmaker_three],
1120         [(4, 8), (6, 8), alto_two_musicmaker_three],
1121         [(6, 8), (8, 8), alto_two_musicmaker_three],
1122
1123         [(8, 8), (10, 8), alto_two_musicmaker_two],
1124         [(10, 8), (12, 8), alto_two_musicmaker_two],
1125         [(12, 8), (14, 8), alto_two_musicmaker_two],
1126         [(14, 8), (16, 8), alto_two_musicmaker_two],
1127
1128         [(16, 8), (18, 8), alto_two_musicmaker_two],
1129         [(18, 8), (20, 8), alto_two_musicmaker_three],
1130         [(20, 8), (22, 8), alto_two_musicmaker_three],
1131         [(22, 8), (24, 8), alto_two_musicmaker_three],
1132
1133         [(24, 8), (26, 8), alto_two_musicmaker_two],
1134         [(26, 8), (28, 8), alto_two_musicmaker_three],
1135         [(28, 8), (30, 8), alto_two_musicmaker_three],
1136         [(30, 8), (32, 8), alto_two_musicmaker_two],
1137
1138         [(32, 8), (34, 8), alto_two_musicmaker_two],
1139         [(34, 8), (36, 8), alto_two_musicmaker_two],
1140         [(36, 8), (38, 8), alto_two_musicmaker_two],
1141         [(38, 8), (40, 8), alto_two_musicmaker_three],
1142
1143         [(40, 8), (42, 8), alto_two_musicmaker_three],
1144         [(42, 8), (44, 8), alto_two_musicmaker_three],
1145         [(44, 8), (46, 8), alto_two_musicmaker_two],
1146         [(46, 8), (48, 8), alto_two_musicmaker_two],
1147
1148         [(48, 8), (50, 8), alto_two_musicmaker_three],
1149         [(50, 8), (52, 8), alto_two_musicmaker_two],
1150         [(52, 8), (54, 8), alto_two_musicmaker_three],
1151         [(54, 8), (56, 8), alto_two_musicmaker_two],
1152
1153         [(56, 8), (58, 8), alto_two_musicmaker_two],
1154         [(58, 8), (60, 8), alto_two_musicmaker_three],
1155         [(60, 8), (62, 8), alto_two_musicmaker_three],
1156         [(62, 8), (64, 8), alto_two_musicmaker_two],
1157

```

```

1158     [(64, 8), (66, 8), alto_two_musicmaker_two],
1159     [(66, 8), (68, 8), alto_two_musicmaker_two],
1160     [(68, 8), (70, 8), alto_two_musicmaker_three],
1161     [(70, 8), (72, 8), alto_two_musicmaker_three],
1162
1163     [(72, 8), (74, 8), alto_two_musicmaker_three],
1164     [(74, 8), (76, 8), alto_two_musicmaker_two],
1165     [(76, 8), (78, 8), alto_two_musicmaker_two],
1166     [(78, 8), (80, 8), alto_two_musicmaker_two],
1167
1168     [(80, 8), (82, 8), alto_two_musicmaker_two],
1169     [(82, 8), (84, 8), alto_two_musicmaker_three],
1170     [(84, 8), (86, 8), alto_two_musicmaker_three],
1171     [(86, 8), (88, 8), alto_two_musicmaker_three],
1172
1173     [(88, 8), (90, 8), alto_two_musicmaker_three],
1174     [(90, 8), (92, 8), alto_two_musicmaker_two],
1175     [(92, 8), (94, 8), alto_two_musicmaker_two],
1176     [(94, 8), (96, 8), alto_two_musicmaker_three],
1177
1178     [(96, 8), (98, 8), alto_two_musicmaker_two],
1179     [(98, 8), (100, 8), alto_two_musicmaker_two],
1180     [(100, 8), (102, 8), alto_two_musicmaker_three],
1181     [(102, 8), (104, 8), alto_two_musicmaker_two],
1182 ]
1183 ])
1184
1185 voice_7_timespan_list = abjad.TimespanList([
1186     abjad.AnnotatedTimespan(
1187         start_offset=start_offset,
1188         stop_offset=stop_offset,
1189         annotation=MusicSpecifier(
1190             music_maker=music_maker,
1191             voice_name='Voice 7',
1192         ),
1193     )
1194     for start_offset, stop_offset, music_maker in [
1195         [(0, 8), (2, 8), alto_three_musicmaker_two],
1196         [(2, 8), (4, 8), alto_three_musicmaker_two],
1197         [(4, 8), (6, 8), alto_three_musicmaker_three],
1198         [(6, 8), (8, 8), alto_three_musicmaker_two],
1199
1200         [(8, 8), (10, 8), alto_three_musicmaker_two],
1201         [(10, 8), (12, 8), alto_three_musicmaker_three],
1202         [(12, 8), (14, 8), alto_three_musicmaker_two],
1203         [(14, 8), (16, 8), alto_three_musicmaker_three],
1204
1205         [(16, 8), (18, 8), alto_three_musicmaker_three],
1206         [(18, 8), (20, 8), alto_three_musicmaker_three],
1207         [(20, 8), (22, 8), alto_three_musicmaker_two],
1208         [(22, 8), (24, 8), alto_three_musicmaker_two],
1209
1210         [(24, 8), (26, 8), alto_three_musicmaker_two],
1211         [(26, 8), (28, 8), alto_three_musicmaker_three],
1212         [(28, 8), (30, 8), alto_three_musicmaker_two],
1213         [(30, 8), (32, 8), alto_three_musicmaker_three],
1214
1215         [(32, 8), (34, 8), alto_three_musicmaker_three],
1216         [(34, 8), (36, 8), alto_three_musicmaker_three],

```

```

1217     [(36, 8), (38, 8), alto_three_musicmaker_two],
1218     [(38, 8), (40, 8), alto_three_musicmaker_three],
1219
1220     [(40, 8), (42, 8), alto_three_musicmaker_three],
1221     [(42, 8), (44, 8), alto_three_musicmaker_two],
1222     [(44, 8), (46, 8), alto_three_musicmaker_three],
1223     [(46, 8), (48, 8), alto_three_musicmaker_three],
1224
1225     [(48, 8), (50, 8), alto_three_musicmaker_three],
1226     [(50, 8), (52, 8), alto_three_musicmaker_two],
1227     [(52, 8), (54, 8), alto_three_musicmaker_three],
1228     [(54, 8), (56, 8), alto_three_musicmaker_two],
1229
1230     [(56, 8), (58, 8), alto_three_musicmaker_two],
1231     [(58, 8), (60, 8), alto_three_musicmaker_three],
1232     [(60, 8), (62, 8), alto_three_musicmaker_three],
1233     [(62, 8), (64, 8), alto_three_musicmaker_three],
1234
1235     [(64, 8), (66, 8), alto_three_musicmaker_two],
1236     [(66, 8), (68, 8), alto_three_musicmaker_two],
1237     [(68, 8), (70, 8), alto_three_musicmaker_two],
1238     [(70, 8), (72, 8), alto_three_musicmaker_two],
1239
1240     [(72, 8), (74, 8), alto_three_musicmaker_two],
1241     [(74, 8), (76, 8), alto_three_musicmaker_three],
1242     [(76, 8), (78, 8), alto_three_musicmaker_three],
1243     [(78, 8), (80, 8), alto_three_musicmaker_three],
1244
1245     [(80, 8), (82, 8), alto_three_musicmaker_three],
1246     [(82, 8), (84, 8), alto_three_musicmaker_two],
1247     [(84, 8), (86, 8), alto_three_musicmaker_three],
1248     [(86, 8), (88, 8), alto_three_musicmaker_two],
1249
1250     [(88, 8), (90, 8), alto_three_musicmaker_two],
1251     [(90, 8), (92, 8), alto_three_musicmaker_two],
1252     [(92, 8), (94, 8), alto_three_musicmaker_three],
1253     [(94, 8), (96, 8), alto_three_musicmaker_two],
1254
1255     [(96, 8), (98, 8), alto_three_musicmaker_two],
1256     [(98, 8), (100, 8), alto_three_musicmaker_three],
1257     [(100, 8), (102, 8), alto_three_musicmaker_three],
1258     [(102, 8), (104, 8), alto_three_musicmaker_three],
1259 ]
1260 ])
1261
1262 voice_8_timespan_list = abjad.TimespanList([
1263     abjad.AnnotatedTimespan(
1264         start_offset=start_offset,
1265         stop_offset=stop_offset,
1266         annotation=MusicSpecifier(
1267             music_maker=music_maker,
1268             voice_name='Voice 8',
1269         ),
1270     )
1271     for start_offset, stop_offset, music_maker in [
1272         [(0, 8), (2, 8), alto_four_musicmaker_two],
1273         [(2, 8), (4, 8), alto_four_musicmaker_two],
1274         [(4, 8), (6, 8), alto_four_musicmaker_two],
1275         [(6, 8), (8, 8), alto_four_musicmaker_three],

```

```
1276  
1277      [(8, 8), (10, 8), alto_four_musicmaker_three],  
1278      [(10, 8), (12, 8), alto_four_musicmaker_three],  
1279      [(12, 8), (14, 8), alto_four_musicmaker_two],  
1280      [(14, 8), (16, 8), alto_four_musicmaker_two],  
1281  
1282      [(16, 8), (18, 8), alto_four_musicmaker_two],  
1283      [(18, 8), (20, 8), alto_four_musicmaker_three],  
1284      [(20, 8), (22, 8), alto_four_musicmaker_three],  
1285      [(22, 8), (24, 8), alto_four_musicmaker_three],  
1286  
1287      [(24, 8), (26, 8), alto_four_musicmaker_three],  
1288      [(26, 8), (28, 8), alto_four_musicmaker_two],  
1289      [(28, 8), (30, 8), alto_four_musicmaker_two],  
1290      [(30, 8), (32, 8), alto_four_musicmaker_two],  
1291  
1292      [(32, 8), (34, 8), alto_four_musicmaker_two],  
1293      [(34, 8), (36, 8), alto_four_musicmaker_three],  
1294      [(36, 8), (38, 8), alto_four_musicmaker_three],  
1295      [(38, 8), (40, 8), alto_four_musicmaker_three],  
1296  
1297      [(40, 8), (42, 8), alto_four_musicmaker_three],  
1298      [(42, 8), (44, 8), alto_four_musicmaker_three],  
1299      [(44, 8), (46, 8), alto_four_musicmaker_two],  
1300      [(46, 8), (48, 8), alto_four_musicmaker_two],  
1301  
1302      [(48, 8), (50, 8), alto_four_musicmaker_two],  
1303      [(50, 8), (52, 8), alto_four_musicmaker_two],  
1304      [(52, 8), (54, 8), alto_four_musicmaker_three],  
1305      [(54, 8), (56, 8), alto_four_musicmaker_three],  
1306  
1307      [(56, 8), (58, 8), alto_four_musicmaker_three],  
1308      [(58, 8), (60, 8), alto_four_musicmaker_two],  
1309      [(60, 8), (62, 8), alto_four_musicmaker_three],  
1310      [(62, 8), (64, 8), alto_four_musicmaker_two],  
1311  
1312      [(64, 8), (66, 8), alto_four_musicmaker_three],  
1313      [(66, 8), (68, 8), alto_four_musicmaker_two],  
1314      [(68, 8), (70, 8), alto_four_musicmaker_three],  
1315      [(70, 8), (72, 8), alto_four_musicmaker_two],  
1316  
1317      [(72, 8), (74, 8), alto_four_musicmaker_two],  
1318      [(74, 8), (76, 8), alto_four_musicmaker_two],  
1319      [(76, 8), (78, 8), alto_four_musicmaker_three],  
1320      [(78, 8), (80, 8), alto_four_musicmaker_three],  
1321  
1322      [(80, 8), (82, 8), alto_four_musicmaker_three],  
1323      [(82, 8), (84, 8), alto_four_musicmaker_two],  
1324      [(84, 8), (86, 8), alto_four_musicmaker_three],  
1325      [(86, 8), (88, 8), alto_four_musicmaker_two],  
1326  
1327      [(88, 8), (90, 8), alto_four_musicmaker_three],  
1328      [(90, 8), (92, 8), alto_four_musicmaker_three],  
1329      [(92, 8), (94, 8), alto_four_musicmaker_two],  
1330      [(94, 8), (96, 8), alto_four_musicmaker_two],  
1331  
1332      [(96, 8), (98, 8), alto_four_musicmaker_two],  
1333      [(98, 8), (100, 8), alto_four_musicmaker_two],  
1334      [(100, 8), (102, 8), alto_four_musicmaker_two],
```

```

1335     ],
1336   ]
1337 ])
1338
1339 voice_9_timespan_list = abjad.TimespanList([
1340     abjad.AnnotatedTimespan(
1341         start_offset=start_offset,
1342         stop_offset=stop_offset,
1343         annotation=MusicSpecifier(
1344             music_maker=music_maker,
1345             voice_name='Voice 9',
1346         ),
1347     )
1348     for start_offset, stop_offset, music_maker in [
1349         [(0, 8), (2, 8), alto_five_musicmaker_two],
1350         [(2, 8), (4, 8), alto_five_musicmaker_three],
1351         [(4, 8), (6, 8), alto_five_musicmaker_three],
1352         [(6, 8), (8, 8), alto_five_musicmaker_two],
1353
1354         [(8, 8), (10, 8), alto_five_musicmaker_two],
1355         [(10, 8), (12, 8), alto_five_musicmaker_two],
1356         [(12, 8), (14, 8), alto_five_musicmaker_three],
1357         [(14, 8), (16, 8), alto_five_musicmaker_two],
1358
1359         [(16, 8), (18, 8), alto_five_musicmaker_three],
1360         [(18, 8), (20, 8), alto_five_musicmaker_three],
1361         [(20, 8), (22, 8), alto_five_musicmaker_two],
1362         [(22, 8), (24, 8), alto_five_musicmaker_three],
1363
1364         [(24, 8), (26, 8), alto_five_musicmaker_two],
1365         [(26, 8), (28, 8), alto_five_musicmaker_two],
1366         [(28, 8), (30, 8), alto_five_musicmaker_three],
1367         [(30, 8), (32, 8), alto_five_musicmaker_two],
1368
1369         [(32, 8), (34, 8), alto_five_musicmaker_three],
1370         [(34, 8), (36, 8), alto_five_musicmaker_three],
1371         [(36, 8), (38, 8), alto_five_musicmaker_two],
1372         [(38, 8), (40, 8), alto_five_musicmaker_three],
1373
1374         [(40, 8), (42, 8), alto_five_musicmaker_three],
1375         [(42, 8), (44, 8), alto_five_musicmaker_two],
1376         [(44, 8), (46, 8), alto_five_musicmaker_three],
1377         [(46, 8), (48, 8), alto_five_musicmaker_two],
1378
1379         [(48, 8), (50, 8), alto_five_musicmaker_two],
1380         [(50, 8), (52, 8), alto_five_musicmaker_two],
1381         [(52, 8), (54, 8), alto_five_musicmaker_three],
1382         [(54, 8), (56, 8), alto_five_musicmaker_two],
1383
1384         [(56, 8), (58, 8), alto_five_musicmaker_three],
1385         [(58, 8), (60, 8), alto_five_musicmaker_three],
1386         [(60, 8), (62, 8), alto_five_musicmaker_three],
1387         [(62, 8), (64, 8), alto_five_musicmaker_two],
1388
1389         [(64, 8), (66, 8), alto_five_musicmaker_two],
1390         [(66, 8), (68, 8), alto_five_musicmaker_three],
1391         [(68, 8), (70, 8), alto_five_musicmaker_three],
1392         [(70, 8), (72, 8), alto_five_musicmaker_three],
1393

```

```

1394     [(72, 8), (74, 8), alto_five_musicmaker_two],
1395     [(74, 8), (76, 8), alto_five_musicmaker_two],
1396     [(76, 8), (78, 8), alto_five_musicmaker_three],
1397     [(78, 8), (80, 8), alto_five_musicmaker_three],
1398
1399     [(80, 8), (82, 8), alto_five_musicmaker_two],
1400     [(82, 8), (84, 8), alto_five_musicmaker_three],
1401     [(84, 8), (86, 8), alto_five_musicmaker_two],
1402     [(86, 8), (88, 8), alto_five_musicmaker_three],
1403
1404     [(88, 8), (90, 8), alto_five_musicmaker_two],
1405     [(90, 8), (92, 8), alto_five_musicmaker_three],
1406     [(92, 8), (94, 8), alto_five_musicmaker_two],
1407     [(94, 8), (96, 8), alto_five_musicmaker_three],
1408
1409     [(96, 8), (98, 8), alto_five_musicmaker_three],
1410     [(98, 8), (100, 8), alto_five_musicmaker_three],
1411     [(100, 8), (102, 8), alto_five_musicmaker_two],
1412     [(102, 8), (104, 8), alto_five_musicmaker_two],
1413 ]
1414 ])
1415
1416 voice_10_timespan_list = abjad.TimespanList([
1417     abjad.AnnotatedTimespan(
1418         start_offset=start_offset,
1419         stop_offset=stop_offset,
1420         annotation=MusicSpecifier(
1421             music_maker=music_maker,
1422             voice_name='Voice 10',
1423         ),
1424     )
1425     for start_offset, stop_offset, music_maker in [
1426         [(0, 8), (2, 8), alto_six_musicmaker_three],
1427         [(2, 8), (4, 8), alto_six_musicmaker_three],
1428         [(4, 8), (6, 8), alto_six_musicmaker_two],
1429         [(6, 8), (8, 8), alto_six_musicmaker_three],
1430
1431         [(8, 8), (10, 8), alto_six_musicmaker_two],
1432         [(10, 8), (12, 8), alto_six_musicmaker_two],
1433         [(12, 8), (14, 8), alto_six_musicmaker_three],
1434         [(14, 8), (16, 8), alto_six_musicmaker_two],
1435
1436         [(16, 8), (18, 8), alto_six_musicmaker_three],
1437         [(18, 8), (20, 8), alto_six_musicmaker_two],
1438         [(20, 8), (22, 8), alto_six_musicmaker_three],
1439         [(22, 8), (24, 8), alto_six_musicmaker_two],
1440
1441         [(24, 8), (26, 8), alto_six_musicmaker_three],
1442         [(26, 8), (28, 8), alto_six_musicmaker_two],
1443         [(28, 8), (30, 8), alto_six_musicmaker_three],
1444         [(30, 8), (32, 8), alto_six_musicmaker_two],
1445
1446         [(32, 8), (34, 8), alto_six_musicmaker_two],
1447         [(34, 8), (36, 8), alto_six_musicmaker_two],
1448         [(36, 8), (38, 8), alto_six_musicmaker_three],
1449         [(38, 8), (40, 8), alto_six_musicmaker_three],
1450
1451         [(40, 8), (42, 8), alto_six_musicmaker_three],
1452         [(42, 8), (44, 8), alto_six_musicmaker_two],

```

```

1453     [(44, 8), (46, 8), alto_six_musicmaker_three],
1454     [(46, 8), (48, 8), alto_six_musicmaker_three],
1455
1456     [(48, 8), (50, 8), alto_six_musicmaker_three],
1457     [(50, 8), (52, 8), alto_six_musicmaker_two],
1458     [(52, 8), (54, 8), alto_six_musicmaker_two],
1459     [(54, 8), (56, 8), alto_six_musicmaker_two],
1460
1461     [(56, 8), (58, 8), alto_six_musicmaker_two],
1462     [(58, 8), (60, 8), alto_six_musicmaker_two],
1463     [(60, 8), (62, 8), alto_six_musicmaker_three],
1464     [(62, 8), (64, 8), alto_six_musicmaker_two],
1465
1466     [(64, 8), (66, 8), alto_six_musicmaker_two],
1467     [(66, 8), (68, 8), alto_six_musicmaker_three],
1468     [(68, 8), (70, 8), alto_six_musicmaker_two],
1469     [(70, 8), (72, 8), alto_six_musicmaker_three],
1470
1471     [(72, 8), (74, 8), alto_six_musicmaker_three],
1472     [(74, 8), (76, 8), alto_six_musicmaker_two],
1473     [(76, 8), (78, 8), alto_six_musicmaker_three],
1474     [(78, 8), (80, 8), alto_six_musicmaker_two],
1475
1476     [(80, 8), (82, 8), alto_six_musicmaker_two],
1477     [(82, 8), (84, 8), alto_six_musicmaker_two],
1478     [(84, 8), (86, 8), alto_six_musicmaker_three],
1479     [(86, 8), (88, 8), alto_six_musicmaker_two],
1480
1481     [(88, 8), (90, 8), alto_six_musicmaker_three],
1482     [(90, 8), (92, 8), alto_six_musicmaker_three],
1483     [(92, 8), (94, 8), alto_six_musicmaker_two],
1484     [(94, 8), (96, 8), alto_six_musicmaker_three],
1485
1486     [(96, 8), (98, 8), alto_six_musicmaker_two],
1487     [(98, 8), (100, 8), alto_six_musicmaker_two],
1488     [(100, 8), (102, 8), alto_six_musicmaker_two],
1489     [(102, 8), (104, 8), alto_six_musicmaker_three],
1490 ]
1491 ])
1492
1493 voice_11_timespan_list = abjad.TimespanList([
1494     abjad.AnnotatedTimespan(
1495         start_offset=start_offset,
1496         stop_offset=stop_offset,
1497         annotation=MusicSpecifier(
1498             music_maker=music_maker,
1499             voice_name='Voice 11',
1500         ),
1501     )
1502     for start_offset, stop_offset, music_maker in [
1503         [(0, 8), (2, 8), tenor_one_musicmaker_three],
1504         [(2, 8), (4, 8), tenor_one_musicmaker_three],
1505         [(4, 8), (6, 8), tenor_one_musicmaker_two],
1506         [(6, 8), (8, 8), tenor_one_musicmaker_two],
1507         [(8, 8), (10, 8), tenor_one_musicmaker_two],
1508         [(10, 8), (12, 8), tenor_one_musicmaker_three],
1509         [(12, 8), (14, 8), tenor_one_musicmaker_three],
1510         [(14, 8), (16, 8), tenor_one_musicmaker_three],
1511         [(16, 8), (18, 8), tenor_one_musicmaker_three],

```

```

1512     [(18, 8), (20, 8), tenor_one_musicmaker_two],
1513     [(20, 8), (22, 8), tenor_one_musicmaker_two],
1514     [(22, 8), (24, 8), tenor_one_musicmaker_two],
1515     [(24, 8), (26, 8), tenor_one_musicmaker_two],
1516     [(26, 8), (28, 8), tenor_one_musicmaker_three],
1517     [(28, 8), (30, 8), tenor_one_musicmaker_three],
1518     [(30, 8), (32, 8), tenor_one_musicmaker_three],
1519     [(32, 8), (34, 8), tenor_one_musicmaker_three],
1520     [(34, 8), (36, 8), tenor_one_musicmaker_three],
1521     [(36, 8), (38, 8), tenor_one_musicmaker_two],
1522     [(38, 8), (40, 8), tenor_one_musicmaker_two],
1523     [(40, 8), (42, 8), tenor_one_musicmaker_two],
1524     [(42, 8), (44, 8), tenor_one_musicmaker_two],
1525     [(44, 8), (46, 8), tenor_one_musicmaker_three],
1526     [(46, 8), (48, 8), tenor_one_musicmaker_three],
1527     [(48, 8), (50, 8), tenor_one_musicmaker_three],
1528     [(50, 8), (52, 8), tenor_one_musicmaker_three],
1529     [(52, 8), (54, 8), tenor_one_musicmaker_two],
1530     [(54, 8), (56, 8), tenor_one_musicmaker_two],
1531     [(56, 8), (58, 8), tenor_one_musicmaker_two],
1532     [(58, 8), (60, 8), tenor_one_musicmaker_three],
1533     [(60, 8), (62, 8), tenor_one_musicmaker_three],
1534     [(62, 8), (64, 8), tenor_one_musicmaker_three],
1535     [(64, 8), (66, 8), tenor_one_musicmaker_two],
1536     [(66, 8), (68, 8), tenor_one_musicmaker_two],
1537     [(68, 8), (70, 8), tenor_one_musicmaker_three],
1538     [(70, 8), (72, 8), tenor_one_musicmaker_two],
1539     [(72, 8), (74, 8), tenor_one_musicmaker_two],
1540     [(74, 8), (76, 8), tenor_one_musicmaker_two],
1541     [(76, 8), (78, 8), tenor_one_musicmaker_three],
1542     [(78, 8), (80, 8), tenor_one_musicmaker_three],
1543     [(80, 8), (82, 8), tenor_one_musicmaker_three],
1544     [(82, 8), (84, 8), tenor_one_musicmaker_three],
1545     [(84, 8), (86, 8), tenor_one_musicmaker_two],
1546     [(86, 8), (88, 8), tenor_one_musicmaker_two],
1547     [(88, 8), (90, 8), tenor_one_musicmaker_two],
1548     [(90, 8), (92, 8), tenor_one_musicmaker_two],
1549     [(92, 8), (94, 8), tenor_one_musicmaker_two],
1550     [(94, 8), (96, 8), tenor_one_musicmaker_two],
1551     [(96, 8), (98, 8), tenor_one_musicmaker_two],
1552     [(98, 8), (100, 8), tenor_one_musicmaker_three],
1553     [(100, 8), (102, 8), tenor_one_musicmaker_three],
1554     [(102, 8), (104, 8), tenor_one_musicmaker_three],
1555   ],
1556 ]
1557 ]
1558 voice_12_timespan_list = abjad.TimespanList([
1559     abjad.AnnotatedTimespan(
1560         start_offset=start_offset,
1561         stop_offset=stop_offset,
1562         annotation=MusicSpecifier(
1563             music_maker=music_maker,
1564             voice_name='Voice 12',
1565         ),
1566     )
1567     for start_offset, stop_offset, music_maker in [
1568         [(0, 8), (2, 8), tenor_two_musicmaker_two],
1569         [(2, 8), (4, 8), tenor_two_musicmaker_three],
1570         [(4, 8), (6, 8), tenor_two_musicmaker_three],

```

```

1571     [(6, 8), (8, 8), tenor_two_musicmaker_three],
1572     [(8, 8), (10, 8), tenor_two_musicmaker_three],
1573     [(10, 8), (12, 8), tenor_two_musicmaker_three],
1574     [(12, 8), (14, 8), tenor_two_musicmaker_three],
1575     [(14, 8), (16, 8), tenor_two_musicmaker_two],
1576     [(16, 8), (18, 8), tenor_two_musicmaker_two],
1577     [(18, 8), (20, 8), tenor_two_musicmaker_two],
1578     [(20, 8), (22, 8), tenor_two_musicmaker_two],
1579     [(22, 8), (24, 8), tenor_two_musicmaker_two],
1580     [(24, 8), (26, 8), tenor_two_musicmaker_two],
1581     [(26, 8), (28, 8), tenor_two_musicmaker_two],
1582     [(28, 8), (30, 8), tenor_two_musicmaker_three],
1583     [(30, 8), (32, 8), tenor_two_musicmaker_three],
1584     [(32, 8), (34, 8), tenor_two_musicmaker_three],
1585     [(34, 8), (36, 8), tenor_two_musicmaker_three],
1586     [(36, 8), (38, 8), tenor_two_musicmaker_three],
1587     [(38, 8), (40, 8), tenor_two_musicmaker_three],
1588     [(40, 8), (42, 8), tenor_two_musicmaker_three],
1589     [(42, 8), (44, 8), tenor_two_musicmaker_two],
1590     [(44, 8), (46, 8), tenor_two_musicmaker_two],
1591     [(46, 8), (48, 8), tenor_two_musicmaker_two],
1592     [(48, 8), (50, 8), tenor_two_musicmaker_three],
1593     [(50, 8), (52, 8), tenor_two_musicmaker_three],
1594     [(52, 8), (54, 8), tenor_two_musicmaker_two],
1595     [(54, 8), (56, 8), tenor_two_musicmaker_two],
1596     [(56, 8), (58, 8), tenor_two_musicmaker_two],
1597     [(58, 8), (60, 8), tenor_two_musicmaker_two],
1598     [(60, 8), (62, 8), tenor_two_musicmaker_two],
1599     [(62, 8), (64, 8), tenor_two_musicmaker_three],
1600     [(64, 8), (66, 8), tenor_two_musicmaker_three],
1601     [(66, 8), (68, 8), tenor_two_musicmaker_three],
1602     [(68, 8), (70, 8), tenor_two_musicmaker_three],
1603     [(70, 8), (72, 8), tenor_two_musicmaker_three],
1604     [(72, 8), (74, 8), tenor_two_musicmaker_two],
1605     [(74, 8), (76, 8), tenor_two_musicmaker_two],
1606     [(76, 8), (78, 8), tenor_two_musicmaker_two],
1607     [(78, 8), (80, 8), tenor_two_musicmaker_two],
1608     [(80, 8), (82, 8), tenor_two_musicmaker_two],
1609     [(82, 8), (84, 8), tenor_two_musicmaker_two],
1610     [(84, 8), (86, 8), tenor_two_musicmaker_two],
1611     [(86, 8), (88, 8), tenor_two_musicmaker_three],
1612     [(88, 8), (90, 8), tenor_two_musicmaker_three],
1613     [(90, 8), (92, 8), tenor_two_musicmaker_three],
1614     [(92, 8), (94, 8), tenor_two_musicmaker_three],
1615     [(94, 8), (96, 8), tenor_two_musicmaker_three],
1616     [(96, 8), (98, 8), tenor_two_musicmaker_three],
1617     [(98, 8), (100, 8), tenor_two_musicmaker_two],
1618     [(100, 8), (102, 8), tenor_two_musicmaker_two],
1619     [(102, 8), (104, 8), tenor_two_musicmaker_two],
1620 ]
1621 ])
1622
1623 voice_13_timespan_list = abjad.TimespanList([
1624     abjad.AnnotatedTimespan(
1625         start_offset=start_offset,
1626         stop_offset=stop_offset,
1627         annotation=MusicSpecifier(
1628             music_maker=music_maker,
1629             voice_name='Voice 13',

```

```

1630     ),
1631 )
1632 for start_offset, stop_offset, music_maker in [
1633     [(0, 8), (2, 8), tenor_three_musicmaker_three],
1634     [(2, 8), (4, 8), tenor_three_musicmaker_three],
1635     [(4, 8), (6, 8), tenor_three_musicmaker_three],
1636     [(6, 8), (8, 8), tenor_three_musicmaker_three],
1637     [(8, 8), (10, 8), tenor_three_musicmaker_three],
1638     [(10, 8), (12, 8), tenor_three_musicmaker_three],
1639     [(12, 8), (14, 8), tenor_three_musicmaker_two],
1640     [(14, 8), (16, 8), tenor_three_musicmaker_two],
1641     [(16, 8), (18, 8), tenor_three_musicmaker_two],
1642     [(18, 8), (20, 8), tenor_three_musicmaker_two],
1643     [(20, 8), (22, 8), tenor_three_musicmaker_two],
1644     [(22, 8), (24, 8), tenor_three_musicmaker_two],
1645     [(24, 8), (26, 8), tenor_three_musicmaker_three],
1646     [(26, 8), (28, 8), tenor_three_musicmaker_three],
1647     [(28, 8), (30, 8), tenor_three_musicmaker_three],
1648     [(30, 8), (32, 8), tenor_three_musicmaker_two],
1649     [(32, 8), (34, 8), tenor_three_musicmaker_two],
1650     [(34, 8), (36, 8), tenor_three_musicmaker_two],
1651     [(36, 8), (38, 8), tenor_three_musicmaker_two],
1652     [(38, 8), (40, 8), tenor_three_musicmaker_three],
1653     [(40, 8), (42, 8), tenor_three_musicmaker_three],
1654     [(42, 8), (44, 8), tenor_three_musicmaker_three],
1655     [(44, 8), (46, 8), tenor_three_musicmaker_three],
1656     [(46, 8), (48, 8), tenor_three_musicmaker_three],
1657     [(48, 8), (50, 8), tenor_three_musicmaker_three],
1658     [(50, 8), (52, 8), tenor_three_musicmaker_three],
1659     [(52, 8), (54, 8), tenor_three_musicmaker_two],
1660     [(54, 8), (56, 8), tenor_three_musicmaker_two],
1661     [(56, 8), (58, 8), tenor_three_musicmaker_two],
1662     [(58, 8), (60, 8), tenor_three_musicmaker_two],
1663     [(60, 8), (62, 8), tenor_three_musicmaker_two],
1664     [(62, 8), (64, 8), tenor_three_musicmaker_three],
1665     [(64, 8), (66, 8), tenor_three_musicmaker_three],
1666     [(66, 8), (68, 8), tenor_three_musicmaker_three],
1667     [(68, 8), (70, 8), tenor_three_musicmaker_three],
1668     [(70, 8), (72, 8), tenor_three_musicmaker_three],
1669     [(72, 8), (74, 8), tenor_three_musicmaker_two],
1670     [(74, 8), (76, 8), tenor_three_musicmaker_two],
1671     [(76, 8), (78, 8), tenor_three_musicmaker_two],
1672     [(78, 8), (80, 8), tenor_three_musicmaker_two],
1673     [(80, 8), (82, 8), tenor_three_musicmaker_two],
1674     [(82, 8), (84, 8), tenor_three_musicmaker_two],
1675     [(84, 8), (86, 8), tenor_three_musicmaker_two],
1676     [(86, 8), (88, 8), tenor_three_musicmaker_three],
1677     [(88, 8), (90, 8), tenor_three_musicmaker_three],
1678     [(90, 8), (92, 8), tenor_three_musicmaker_three],
1679     [(92, 8), (94, 8), tenor_three_musicmaker_three],
1680     [(94, 8), (96, 8), tenor_three_musicmaker_two],
1681     [(96, 8), (98, 8), tenor_three_musicmaker_two],
1682     [(98, 8), (100, 8), tenor_three_musicmaker_two],
1683     [(100, 8), (102, 8), tenor_three_musicmaker_two],
1684     [(102, 8), (104, 8), tenor_three_musicmaker_two],
1685   ]
1686 ])
1687
1688 voice_14_timespan_list = abjad.TimespanList([

```

```

1689 abjad.AnnotatedTimespan(
1690     start_offset=start_offset,
1691     stop_offset=stop_offset,
1692     annotation=MusicSpecifier(
1693         music_maker=music_maker,
1694         voice_name='Voice 14',
1695     ),
1696 )
1697 for start_offset, stop_offset, music_maker in [
1698     [(0, 8), (2, 8), tenor_four_musicmaker_two],
1699     [(2, 8), (4, 8), tenor_four_musicmaker_two],
1700     [(4, 8), (6, 8), tenor_four_musicmaker_two],
1701     [(6, 8), (8, 8), tenor_four_musicmaker_two],
1702     [(8, 8), (10, 8), tenor_four_musicmaker_two],
1703     [(10, 8), (12, 8), tenor_four_musicmaker_two],
1704     [(12, 8), (14, 8), tenor_four_musicmaker_two],
1705     [(14, 8), (16, 8), tenor_four_musicmaker_two],
1706     [(16, 8), (18, 8), tenor_four_musicmaker_three],
1707     [(18, 8), (20, 8), tenor_four_musicmaker_three],
1708     [(20, 8), (22, 8), tenor_four_musicmaker_three],
1709     [(22, 8), (24, 8), tenor_four_musicmaker_three],
1710     [(24, 8), (26, 8), tenor_four_musicmaker_three],
1711     [(26, 8), (28, 8), tenor_four_musicmaker_three],
1712     [(28, 8), (30, 8), tenor_four_musicmaker_three],
1713     [(30, 8), (32, 8), tenor_four_musicmaker_three],
1714     [(32, 8), (34, 8), tenor_four_musicmaker_two],
1715     [(34, 8), (36, 8), tenor_four_musicmaker_two],
1716     [(36, 8), (38, 8), tenor_four_musicmaker_two],
1717     [(38, 8), (40, 8), tenor_four_musicmaker_two],
1718     [(40, 8), (42, 8), tenor_four_musicmaker_two],
1719     [(42, 8), (44, 8), tenor_four_musicmaker_two],
1720     [(44, 8), (46, 8), tenor_four_musicmaker_two],
1721     [(46, 8), (48, 8), tenor_four_musicmaker_two],
1722     [(48, 8), (50, 8), tenor_four_musicmaker_three],
1723     [(50, 8), (52, 8), tenor_four_musicmaker_three],
1724     [(52, 8), (54, 8), tenor_four_musicmaker_three],
1725     [(54, 8), (56, 8), tenor_four_musicmaker_three],
1726     [(56, 8), (58, 8), tenor_four_musicmaker_three],
1727     [(58, 8), (60, 8), tenor_four_musicmaker_three],
1728     [(60, 8), (62, 8), tenor_four_musicmaker_three],
1729     [(62, 8), (64, 8), tenor_four_musicmaker_two],
1730     [(64, 8), (66, 8), tenor_four_musicmaker_two],
1731     [(66, 8), (68, 8), tenor_four_musicmaker_two],
1732     [(68, 8), (70, 8), tenor_four_musicmaker_two],
1733     [(70, 8), (72, 8), tenor_four_musicmaker_two],
1734     [(72, 8), (74, 8), tenor_four_musicmaker_two],
1735     [(74, 8), (76, 8), tenor_four_musicmaker_three],
1736     [(76, 8), (78, 8), tenor_four_musicmaker_three],
1737     [(78, 8), (80, 8), tenor_four_musicmaker_three],
1738     [(80, 8), (82, 8), tenor_four_musicmaker_three],
1739     [(82, 8), (84, 8), tenor_four_musicmaker_three],
1740     [(84, 8), (86, 8), tenor_four_musicmaker_two],
1741     [(86, 8), (88, 8), tenor_four_musicmaker_two],
1742     [(88, 8), (90, 8), tenor_four_musicmaker_two],
1743     [(90, 8), (92, 8), tenor_four_musicmaker_two],
1744     [(92, 8), (94, 8), tenor_four_musicmaker_three],
1745     [(94, 8), (96, 8), tenor_four_musicmaker_three],
1746     [(96, 8), (98, 8), tenor_four_musicmaker_three],
1747     [(98, 8), (100, 8), tenor_four_musicmaker_two],

```

```

1748     [(100, 8), (102, 8), tenor_four_musicmaker_two],
1749     [(102, 8), (104, 8), tenor_four_musicmaker_three],
1750   ]
1751 ])
1752
1753 voice_15_timespan_list = abjad.TimespanList([
1754     abjad.AnnotatedTimespan(
1755         start_offset=start_offset,
1756         stop_offset=stop_offset,
1757         annotation=MusicSpecifier(
1758             music_maker=music_maker,
1759             voice_name='Voice 15',
1760         ),
1761     ),
1762     for start_offset, stop_offset, music_maker in [
1763         [(0, 8), (2, 8), tenor_five_musicmaker_two],
1764         [(2, 8), (4, 8), tenor_five_musicmaker_two],
1765         [(4, 8), (6, 8), tenor_five_musicmaker_two],
1766         [(6, 8), (8, 8), tenor_five_musicmaker_three],
1767         [(8, 8), (10, 8), tenor_five_musicmaker_three],
1768         [(10, 8), (12, 8), tenor_five_musicmaker_three],
1769         [(12, 8), (14, 8), tenor_five_musicmaker_two],
1770         [(14, 8), (16, 8), tenor_five_musicmaker_two],
1771         [(16, 8), (18, 8), tenor_five_musicmaker_two],
1772         [(18, 8), (20, 8), tenor_five_musicmaker_two],
1773         [(20, 8), (22, 8), tenor_five_musicmaker_three],
1774         [(22, 8), (24, 8), tenor_five_musicmaker_three],
1775         [(24, 8), (26, 8), tenor_five_musicmaker_three],
1776         [(26, 8), (28, 8), tenor_five_musicmaker_three],
1777         [(28, 8), (30, 8), tenor_five_musicmaker_two],
1778         [(30, 8), (32, 8), tenor_five_musicmaker_two],
1779         [(32, 8), (34, 8), tenor_five_musicmaker_two],
1780         [(34, 8), (36, 8), tenor_five_musicmaker_three],
1781         [(36, 8), (38, 8), tenor_five_musicmaker_three],
1782         [(38, 8), (40, 8), tenor_five_musicmaker_three],
1783         [(40, 8), (42, 8), tenor_five_musicmaker_three],
1784         [(42, 8), (44, 8), tenor_five_musicmaker_two],
1785         [(44, 8), (46, 8), tenor_five_musicmaker_two],
1786         [(46, 8), (48, 8), tenor_five_musicmaker_three],
1787         [(48, 8), (50, 8), tenor_five_musicmaker_three],
1788         [(50, 8), (52, 8), tenor_five_musicmaker_three],
1789         [(52, 8), (54, 8), tenor_five_musicmaker_three],
1790         [(54, 8), (56, 8), tenor_five_musicmaker_three],
1791         [(56, 8), (58, 8), tenor_five_musicmaker_three],
1792         [(58, 8), (60, 8), tenor_five_musicmaker_two],
1793         [(60, 8), (62, 8), tenor_five_musicmaker_three],
1794         [(62, 8), (64, 8), tenor_five_musicmaker_three],
1795         [(64, 8), (66, 8), tenor_five_musicmaker_three],
1796         [(66, 8), (68, 8), tenor_five_musicmaker_three],
1797         [(68, 8), (70, 8), tenor_five_musicmaker_three],
1798         [(70, 8), (72, 8), tenor_five_musicmaker_three],
1799         [(72, 8), (74, 8), tenor_five_musicmaker_three],
1800         [(74, 8), (76, 8), tenor_five_musicmaker_two],
1801         [(76, 8), (78, 8), tenor_five_musicmaker_two],
1802         [(78, 8), (80, 8), tenor_five_musicmaker_two],
1803         [(80, 8), (82, 8), tenor_five_musicmaker_two],
1804         [(82, 8), (84, 8), tenor_five_musicmaker_two],
1805         [(84, 8), (86, 8), tenor_five_musicmaker_two],
1806         [(86, 8), (88, 8), tenor_five_musicmaker_two],

```

```

1807     [(88, 8), (90, 8), tenor_five_musicmaker_two],
1808     [(90, 8), (92, 8), tenor_five_musicmaker_three],
1809     [(92, 8), (94, 8), tenor_five_musicmaker_three],
1810     [(94, 8), (96, 8), tenor_five_musicmaker_three],
1811     [(96, 8), (98, 8), tenor_five_musicmaker_three],
1812     [(98, 8), (100, 8), tenor_five_musicmaker_three],
1813     [(100, 8), (102, 8), tenor_five_musicmaker_three],
1814     [(102, 8), (104, 8), tenor_five_musicmaker_three],
1815 ]
1816 ])
1817
1818 voice_16_timespan_list = abjad.TimespanList([
1819     abjad.AnnotatedTimespan(
1820         start_offset=start_offset,
1821         stop_offset=stop_offset,
1822         annotation=MusicSpecifier(
1823             music_maker=music_maker,
1824             voice_name='Voice 16',
1825         ),
1826     )
1827     for start_offset, stop_offset, music_maker in [
1828         [(0, 8), (2, 8), baritone_one_musicmaker_two],
1829         [(2, 8), (4, 8), baritone_one_musicmaker_two],
1830         [(4, 8), (6, 8), baritone_one_musicmaker_two],
1831         [(6, 8), (8, 8), baritone_one_musicmaker_two],
1832         [(8, 8), (10, 8), baritone_one_musicmaker_three],
1833         [(10, 8), (12, 8), baritone_one_musicmaker_three],
1834         [(12, 8), (14, 8), baritone_one_musicmaker_three],
1835         [(14, 8), (16, 8), baritone_one_musicmaker_two],
1836         [(16, 8), (18, 8), baritone_one_musicmaker_two],
1837         [(18, 8), (20, 8), baritone_one_musicmaker_two],
1838         [(20, 8), (22, 8), baritone_one_musicmaker_two],
1839         [(22, 8), (24, 8), baritone_one_musicmaker_three],
1840         [(24, 8), (26, 8), baritone_one_musicmaker_three],
1841         [(26, 8), (28, 8), baritone_one_musicmaker_three],
1842         [(28, 8), (30, 8), baritone_one_musicmaker_two],
1843         [(30, 8), (32, 8), baritone_one_musicmaker_two],
1844         [(32, 8), (34, 8), baritone_one_musicmaker_two],
1845         [(34, 8), (36, 8), baritone_one_musicmaker_two],
1846         [(36, 8), (38, 8), baritone_one_musicmaker_three],
1847         [(38, 8), (40, 8), baritone_one_musicmaker_three],
1848         [(40, 8), (42, 8), baritone_one_musicmaker_three],
1849         [(42, 8), (44, 8), baritone_one_musicmaker_two],
1850         [(44, 8), (46, 8), baritone_one_musicmaker_two],
1851         [(46, 8), (48, 8), baritone_one_musicmaker_two],
1852         [(48, 8), (50, 8), baritone_one_musicmaker_two],
1853         [(50, 8), (52, 8), baritone_one_musicmaker_three],
1854         [(52, 8), (54, 8), baritone_one_musicmaker_three],
1855         [(54, 8), (56, 8), baritone_one_musicmaker_three],
1856         [(56, 8), (58, 8), baritone_one_musicmaker_two],
1857         [(58, 8), (60, 8), baritone_one_musicmaker_two],
1858         [(60, 8), (62, 8), baritone_one_musicmaker_two],
1859         [(62, 8), (64, 8), baritone_one_musicmaker_two],
1860         [(64, 8), (66, 8), baritone_one_musicmaker_three],
1861         [(66, 8), (68, 8), baritone_one_musicmaker_three],
1862         [(68, 8), (70, 8), baritone_one_musicmaker_three],
1863         [(70, 8), (72, 8), baritone_one_musicmaker_two],
1864         [(72, 8), (74, 8), baritone_one_musicmaker_two],
1865         [(74, 8), (76, 8), baritone_one_musicmaker_two],

```

```

1866     [(76, 8), (78, 8), baritone_one_musicmaker_two],
1867     [(78, 8), (80, 8), baritone_one_musicmaker_three],
1868     [(80, 8), (82, 8), baritone_one_musicmaker_three],
1869     [(82, 8), (84, 8), baritone_one_musicmaker_three],
1870     [(84, 8), (86, 8), baritone_one_musicmaker_two],
1871     [(86, 8), (88, 8), baritone_one_musicmaker_two],
1872     [(88, 8), (90, 8), baritone_one_musicmaker_two],
1873     [(90, 8), (92, 8), baritone_one_musicmaker_two],
1874     [(92, 8), (94, 8), baritone_one_musicmaker_three],
1875     [(94, 8), (96, 8), baritone_one_musicmaker_three],
1876     [(96, 8), (98, 8), baritone_one_musicmaker_three],
1877     [(98, 8), (100, 8), baritone_one_musicmaker_two],
1878     [(100, 8), (102, 8), baritone_one_musicmaker_two],
1879     [(102, 8), (104, 8), baritone_one_musicmaker_two],
1880 ]
1881 ])
1882
1883 voice_17_timespan_list = abjad.TimespanList([
1884     abjad.AnnotatedTimespan(
1885         start_offset=start_offset,
1886         stop_offset=stop_offset,
1887         annotation=MusicSpecifier(
1888             music_maker=music_maker,
1889             voice_name='Voice 17',
1890         ),
1891     ),
1892     for start_offset, stop_offset, music_maker in [
1893         [(0, 8), (2, 8), baritone_two_musicmaker_three],
1894         [(2, 8), (4, 8), baritone_two_musicmaker_three],
1895         [(4, 8), (6, 8), baritone_two_musicmaker_two],
1896         [(6, 8), (8, 8), baritone_two_musicmaker_two],
1897         [(8, 8), (10, 8), baritone_two_musicmaker_two],
1898         [(10, 8), (12, 8), baritone_two_musicmaker_three],
1899         [(12, 8), (14, 8), baritone_two_musicmaker_three],
1900         [(14, 8), (16, 8), baritone_two_musicmaker_three],
1901         [(16, 8), (18, 8), baritone_two_musicmaker_three],
1902         [(18, 8), (20, 8), baritone_two_musicmaker_two],
1903         [(20, 8), (22, 8), baritone_two_musicmaker_two],
1904         [(22, 8), (24, 8), baritone_two_musicmaker_two],
1905         [(24, 8), (26, 8), baritone_two_musicmaker_two],
1906         [(26, 8), (28, 8), baritone_two_musicmaker_two],
1907         [(28, 8), (30, 8), baritone_two_musicmaker_three],
1908         [(30, 8), (32, 8), baritone_two_musicmaker_three],
1909         [(32, 8), (34, 8), baritone_two_musicmaker_three],
1910         [(34, 8), (36, 8), baritone_two_musicmaker_three],
1911         [(36, 8), (38, 8), baritone_two_musicmaker_three],
1912         [(38, 8), (40, 8), baritone_two_musicmaker_three],
1913         [(40, 8), (42, 8), baritone_two_musicmaker_two],
1914         [(42, 8), (44, 8), baritone_two_musicmaker_two],
1915         [(44, 8), (46, 8), baritone_two_musicmaker_two],
1916         [(46, 8), (48, 8), baritone_two_musicmaker_two],
1917         [(48, 8), (50, 8), baritone_two_musicmaker_two],
1918         [(50, 8), (52, 8), baritone_two_musicmaker_two],
1919         [(52, 8), (54, 8), baritone_two_musicmaker_two],
1920         [(54, 8), (56, 8), baritone_two_musicmaker_three],
1921         [(56, 8), (58, 8), baritone_two_musicmaker_three],
1922         [(58, 8), (60, 8), baritone_two_musicmaker_three],
1923         [(60, 8), (62, 8), baritone_two_musicmaker_three],
1924         [(62, 8), (64, 8), baritone_two_musicmaker_three],

```

```

1925     [(64, 8), (66, 8), baritone_two_musicmaker_three],
1926     [(66, 8), (68, 8), baritone_two_musicmaker_three],
1927     [(68, 8), (70, 8), baritone_two_musicmaker_two],
1928     [(70, 8), (72, 8), baritone_two_musicmaker_two],
1929     [(72, 8), (74, 8), baritone_two_musicmaker_two],
1930     [(74, 8), (76, 8), baritone_two_musicmaker_two],
1931     [(76, 8), (78, 8), baritone_two_musicmaker_two],
1932     [(78, 8), (80, 8), baritone_two_musicmaker_two],
1933     [(80, 8), (82, 8), baritone_two_musicmaker_two],
1934     [(82, 8), (84, 8), baritone_two_musicmaker_three],
1935     [(84, 8), (86, 8), baritone_two_musicmaker_three],
1936     [(86, 8), (88, 8), baritone_two_musicmaker_three],
1937     [(88, 8), (90, 8), baritone_two_musicmaker_three],
1938     [(90, 8), (92, 8), baritone_two_musicmaker_three],
1939     [(92, 8), (94, 8), baritone_two_musicmaker_three],
1940     [(94, 8), (96, 8), baritone_two_musicmaker_three],
1941     [(96, 8), (98, 8), baritone_two_musicmaker_two],
1942     [(98, 8), (100, 8), baritone_two_musicmaker_two],
1943     [(100, 8), (102, 8), baritone_two_musicmaker_two],
1944     [(102, 8), (104, 8), baritone_two_musicmaker_two],
1945   ],
1946 ])
1947
1948 voice_18_timespan_list = abjad.TimespanList([
1949     abjad.AnnotatedTimespan(
1950         start_offset=start_offset,
1951         stop_offset=stop_offset,
1952         annotation=MusicSpecifier(
1953             music_maker=music_maker,
1954             voice_name='Voice 18',
1955         ),
1956     ),
1957     for start_offset, stop_offset, music_maker in [
1958         [(0, 8), (2, 8), baritone_three_musicmaker_three],
1959         [(2, 8), (4, 8), baritone_three_musicmaker_three],
1960         [(4, 8), (6, 8), baritone_three_musicmaker_three],
1961         [(6, 8), (8, 8), baritone_three_musicmaker_three],
1962         [(8, 8), (10, 8), baritone_three_musicmaker_two],
1963         [(10, 8), (12, 8), baritone_three_musicmaker_two],
1964         [(12, 8), (14, 8), baritone_three_musicmaker_two],
1965         [(14, 8), (16, 8), baritone_three_musicmaker_two],
1966         [(16, 8), (18, 8), baritone_three_musicmaker_three],
1967         [(18, 8), (20, 8), baritone_three_musicmaker_three],
1968         [(20, 8), (22, 8), baritone_three_musicmaker_three],
1969         [(22, 8), (24, 8), baritone_three_musicmaker_three],
1970         [(24, 8), (26, 8), baritone_three_musicmaker_two],
1971         [(26, 8), (28, 8), baritone_three_musicmaker_two],
1972         [(28, 8), (30, 8), baritone_three_musicmaker_two],
1973         [(30, 8), (32, 8), baritone_three_musicmaker_two],
1974         [(32, 8), (34, 8), baritone_three_musicmaker_three],
1975         [(34, 8), (36, 8), baritone_three_musicmaker_three],
1976         [(36, 8), (38, 8), baritone_three_musicmaker_three],
1977         [(38, 8), (40, 8), baritone_three_musicmaker_three],
1978         [(40, 8), (42, 8), baritone_three_musicmaker_two],
1979         [(42, 8), (44, 8), baritone_three_musicmaker_two],
1980         [(44, 8), (46, 8), baritone_three_musicmaker_two],
1981         [(46, 8), (48, 8), baritone_three_musicmaker_two],
1982         [(48, 8), (50, 8), baritone_three_musicmaker_three],
1983         [(50, 8), (52, 8), baritone_three_musicmaker_three],

```

```

1984     [(52, 8), (54, 8), baritone_three_musicmaker_three],
1985     [(54, 8), (56, 8), baritone_three_musicmaker_three],
1986     [(56, 8), (58, 8), baritone_three_musicmaker_two],
1987     [(58, 8), (60, 8), baritone_three_musicmaker_two],
1988     [(60, 8), (62, 8), baritone_three_musicmaker_two],
1989     [(62, 8), (64, 8), baritone_three_musicmaker_two],
1990     [(64, 8), (66, 8), baritone_three_musicmaker_three],
1991     [(66, 8), (68, 8), baritone_three_musicmaker_three],
1992     [(68, 8), (70, 8), baritone_three_musicmaker_three],
1993     [(70, 8), (72, 8), baritone_three_musicmaker_three],
1994     [(72, 8), (74, 8), baritone_three_musicmaker_two],
1995     [(74, 8), (76, 8), baritone_three_musicmaker_two],
1996     [(76, 8), (78, 8), baritone_three_musicmaker_two],
1997     [(78, 8), (80, 8), baritone_three_musicmaker_two],
1998     [(80, 8), (82, 8), baritone_three_musicmaker_three],
1999     [(82, 8), (84, 8), baritone_three_musicmaker_three],
2000     [(84, 8), (86, 8), baritone_three_musicmaker_three],
2001     [(86, 8), (88, 8), baritone_three_musicmaker_three],
2002     [(88, 8), (90, 8), baritone_three_musicmaker_two],
2003     [(90, 8), (92, 8), baritone_three_musicmaker_two],
2004     [(92, 8), (94, 8), baritone_three_musicmaker_two],
2005     [(94, 8), (96, 8), baritone_three_musicmaker_two],
2006     [(96, 8), (98, 8), baritone_three_musicmaker_three],
2007     [(98, 8), (100, 8), baritone_three_musicmaker_three],
2008     [(100, 8), (102, 8), baritone_three_musicmaker_three],
2009     [(102, 8), (104, 8), baritone_three_musicmaker_three],
2010   ]
2011 ])
2012
2013 voice_19_timespan_list = abjad.TimespanList([
2014     abjad.AnnotatedTimespan(
2015         start_offset=start_offset,
2016         stop_offset=stop_offset,
2017         annotation=MusicSpecifier(
2018             music_maker=music_maker,
2019             voice_name='Voice 19',
2020         ),
2021     )
2022     for start_offset, stop_offset, music_maker in [
2023         [(0, 8), (2, 8), bass_one_musicmaker_two],
2024         [(2, 8), (4, 8), bass_one_musicmaker_two],
2025         [(4, 8), (6, 8), bass_one_musicmaker_two],
2026         [(6, 8), (8, 8), bass_one_musicmaker_three],
2027         [(8, 8), (10, 8), bass_one_musicmaker_three],
2028         [(10, 8), (12, 8), bass_one_musicmaker_two],
2029         [(12, 8), (14, 8), bass_one_musicmaker_two],
2030         [(14, 8), (16, 8), bass_one_musicmaker_two],
2031         [(16, 8), (18, 8), bass_one_musicmaker_two],
2032         [(18, 8), (20, 8), bass_one_musicmaker_three],
2033         [(20, 8), (22, 8), bass_one_musicmaker_two],
2034         [(22, 8), (24, 8), bass_one_musicmaker_two],
2035         [(24, 8), (26, 8), bass_one_musicmaker_two],
2036         [(26, 8), (28, 8), bass_one_musicmaker_two],
2037         [(28, 8), (30, 8), bass_one_musicmaker_two],
2038         [(30, 8), (32, 8), bass_one_musicmaker_three],
2039         [(32, 8), (34, 8), bass_one_musicmaker_three],
2040         [(34, 8), (36, 8), bass_one_musicmaker_three],
2041         [(36, 8), (38, 8), bass_one_musicmaker_three],
2042         [(38, 8), (40, 8), bass_one_musicmaker_three],

```

```

2043     [(40, 8), (42, 8), bass_one_musicmaker_three],
2044     [(42, 8), (44, 8), bass_one_musicmaker_three],
2045     [(44, 8), (46, 8), bass_one_musicmaker_three],
2046     [(46, 8), (48, 8), bass_one_musicmaker_two],
2047     [(48, 8), (50, 8), bass_one_musicmaker_two],
2048     [(50, 8), (52, 8), bass_one_musicmaker_two],
2049     [(52, 8), (54, 8), bass_one_musicmaker_two],
2050     [(54, 8), (56, 8), bass_one_musicmaker_two],
2051     [(56, 8), (58, 8), bass_one_musicmaker_two],
2052     [(58, 8), (60, 8), bass_one_musicmaker_three],
2053     [(60, 8), (62, 8), bass_one_musicmaker_three],
2054     [(62, 8), (64, 8), bass_one_musicmaker_three],
2055     [(64, 8), (68, 8), bass_one_musicmaker_three],
2056     [(68, 8), (70, 8), bass_one_musicmaker_three],
2057     [(70, 8), (72, 8), bass_one_musicmaker_three],
2058     [(72, 8), (74, 8), bass_one_musicmaker_three],
2059     [(74, 8), (76, 8), bass_one_musicmaker_two],
2060     [(76, 8), (78, 8), bass_one_musicmaker_two],
2061     [(78, 8), (80, 8), bass_one_musicmaker_two],
2062     [(80, 8), (82, 8), bass_one_musicmaker_two],
2063     [(82, 8), (84, 8), bass_one_musicmaker_two],
2064     [(84, 8), (86, 8), bass_one_musicmaker_two],
2065     [(86, 8), (88, 8), bass_one_musicmaker_two],
2066     [(88, 8), (90, 8), bass_one_musicmaker_three],
2067     [(90, 8), (92, 8), bass_one_musicmaker_three],
2068     [(92, 8), (94, 8), bass_one_musicmaker_three],
2069     [(94, 8), (96, 8), bass_one_musicmaker_three],
2070     [(96, 8), (98, 8), bass_one_musicmaker_three],
2071     [(98, 8), (100, 8), bass_one_musicmaker_three],
2072     [(100, 8), (102, 8), bass_one_musicmaker_two],
2073     [(102, 8), (104, 8), bass_one_musicmaker_two],
2074   ],
2075 ]
2076
2077 voice_20_timespan_list = abjad.TimespanList([
2078     abjad.AnnotatedTimespan(
2079         start_offset=start_offset,
2080         stop_offset=stop_offset,
2081         annotation=MusicSpecifier(
2082             music_maker=music_maker,
2083             voice_name='Voice 20',
2084         ),
2085     )
2086     for start_offset, stop_offset, music_maker in [
2087         [(0, 8), (2, 8), bass_two_musicmaker_two],
2088         [(2, 8), (4, 8), bass_two_musicmaker_two],
2089         [(4, 8), (6, 8), bass_two_musicmaker_two],
2090         [(6, 8), (8, 8), bass_two_musicmaker_three],
2091         [(8, 8), (10, 8), bass_two_musicmaker_three],
2092         [(10, 8), (12, 8), bass_two_musicmaker_two],
2093         [(12, 8), (14, 8), bass_two_musicmaker_two],
2094         [(14, 8), (16, 8), bass_two_musicmaker_two],
2095         [(16, 8), (18, 8), bass_two_musicmaker_two],
2096         [(18, 8), (20, 8), bass_two_musicmaker_three],
2097         [(20, 8), (22, 8), bass_two_musicmaker_two],
2098         [(22, 8), (24, 8), bass_two_musicmaker_two],
2099         [(24, 8), (26, 8), bass_two_musicmaker_two],
2100         [(26, 8), (28, 8), bass_two_musicmaker_two],
2101         [(28, 8), (30, 8), bass_two_musicmaker_two],

```

```

2102     [(30, 8), (32, 8), bass_two_musicmaker_three],
2103     [(32, 8), (34, 8), bass_two_musicmaker_three],
2104     [(34, 8), (36, 8), bass_two_musicmaker_three],
2105     [(36, 8), (38, 8), bass_two_musicmaker_three],
2106     [(38, 8), (40, 8), bass_two_musicmaker_three],
2107     [(40, 8), (42, 8), bass_two_musicmaker_three],
2108     [(42, 8), (44, 8), bass_two_musicmaker_three],
2109     [(44, 8), (46, 8), bass_two_musicmaker_three],
2110     [(46, 8), (48, 8), bass_two_musicmaker_two],
2111     [(48, 8), (50, 8), bass_two_musicmaker_two],
2112     [(50, 8), (52, 8), bass_two_musicmaker_two],
2113     [(52, 8), (54, 8), bass_two_musicmaker_two],
2114     [(54, 8), (56, 8), bass_two_musicmaker_two],
2115     [(56, 8), (58, 8), bass_two_musicmaker_two],
2116     [(58, 8), (60, 8), bass_two_musicmaker_three],
2117     [(60, 8), (62, 8), bass_two_musicmaker_three],
2118     [(62, 8), (64, 8), bass_two_musicmaker_three],
2119     [(64, 8), (68, 8), bass_two_musicmaker_three],
2120     [(68, 8), (70, 8), bass_two_musicmaker_three],
2121     [(70, 8), (72, 8), bass_two_musicmaker_three],
2122     [(72, 8), (74, 8), bass_two_musicmaker_three],
2123     [(74, 8), (76, 8), bass_two_musicmaker_two],
2124     [(76, 8), (78, 8), bass_two_musicmaker_two],
2125     [(78, 8), (80, 8), bass_two_musicmaker_two],
2126     [(80, 8), (82, 8), bass_two_musicmaker_two],
2127     [(82, 8), (84, 8), bass_two_musicmaker_two],
2128     [(84, 8), (86, 8), bass_two_musicmaker_two],
2129     [(86, 8), (88, 8), bass_two_musicmaker_two],
2130     [(88, 8), (90, 8), bass_two_musicmaker_three],
2131     [(90, 8), (92, 8), bass_two_musicmaker_three],
2132     [(92, 8), (94, 8), bass_two_musicmaker_three],
2133     [(94, 8), (96, 8), bass_two_musicmaker_three],
2134     [(96, 8), (98, 8), bass_two_musicmaker_three],
2135     [(98, 8), (100, 8), bass_two_musicmaker_three],
2136     [(100, 8), (102, 8), bass_two_musicmaker_two],
2137     [(102, 8), (104, 8), bass_two_musicmaker_two],
2138 ]
2139 ])
2140
2141 voice_21_timespan_list = abjad.TimespanList([
2142     abjad.AnnotatedTimespan(
2143         start_offset=start_offset,
2144         stop_offset=stop_offset,
2145         annotation=MusicSpecifier(
2146             music_maker=music_maker,
2147             voice_name='Voice 21',
2148         ),
2149     )
2150     for start_offset, stop_offset, music_maker in [
2151         [(0, 8), (2, 8), contrabass_musicmaker_two],
2152         [(2, 8), (4, 8), contrabass_musicmaker_two],
2153         [(4, 8), (6, 8), contrabass_musicmaker_two],
2154         [(6, 8), (8, 8), contrabass_musicmaker_three],
2155         [(8, 8), (10, 8), contrabass_musicmaker_three],
2156         [(10, 8), (12, 8), contrabass_musicmaker_two],
2157         [(12, 8), (14, 8), contrabass_musicmaker_two],
2158         [(14, 8), (16, 8), contrabass_musicmaker_two],
2159         [(16, 8), (18, 8), contrabass_musicmaker_two],
2160         [(18, 8), (20, 8), contrabass_musicmaker_three],

```

```

2161     [(20, 8), (22, 8), contrabass_musicmaker_two],
2162     [(22, 8), (24, 8), contrabass_musicmaker_two],
2163     [(24, 8), (26, 8), contrabass_musicmaker_two],
2164     [(26, 8), (28, 8), contrabass_musicmaker_two],
2165     [(28, 8), (30, 8), contrabass_musicmaker_two],
2166     [(30, 8), (32, 8), contrabass_musicmaker_three],
2167     [(32, 8), (34, 8), contrabass_musicmaker_three],
2168     [(34, 8), (36, 8), contrabass_musicmaker_three],
2169     [(36, 8), (38, 8), contrabass_musicmaker_three],
2170     [(38, 8), (40, 8), contrabass_musicmaker_three],
2171     [(40, 8), (42, 8), contrabass_musicmaker_three],
2172     [(42, 8), (44, 8), contrabass_musicmaker_three],
2173     [(44, 8), (46, 8), contrabass_musicmaker_three],
2174     [(46, 8), (48, 8), contrabass_musicmaker_two],
2175     [(48, 8), (50, 8), contrabass_musicmaker_two],
2176     [(50, 8), (52, 8), contrabass_musicmaker_two],
2177     [(52, 8), (54, 8), contrabass_musicmaker_two],
2178     [(54, 8), (56, 8), contrabass_musicmaker_two],
2179     [(56, 8), (58, 8), contrabass_musicmaker_two],
2180     [(58, 8), (60, 8), contrabass_musicmaker_three],
2181     [(60, 8), (62, 8), contrabass_musicmaker_three],
2182     [(62, 8), (64, 8), contrabass_musicmaker_three],
2183     [(64, 8), (68, 8), contrabass_musicmaker_three],
2184     [(68, 8), (70, 8), contrabass_musicmaker_three],
2185     [(70, 8), (72, 8), contrabass_musicmaker_three],
2186     [(72, 8), (74, 8), contrabass_musicmaker_three],
2187     [(74, 8), (76, 8), contrabass_musicmaker_two],
2188     [(76, 8), (78, 8), contrabass_musicmaker_two],
2189     [(78, 8), (80, 8), contrabass_musicmaker_two],
2190     [(80, 8), (82, 8), contrabass_musicmaker_two],
2191     [(82, 8), (84, 8), contrabass_musicmaker_two],
2192     [(84, 8), (86, 8), contrabass_musicmaker_two],
2193     [(86, 8), (88, 8), contrabass_musicmaker_two],
2194     [(88, 8), (90, 8), contrabass_musicmaker_three],
2195     [(90, 8), (92, 8), contrabass_musicmaker_three],
2196     [(92, 8), (94, 8), contrabass_musicmaker_three],
2197     [(94, 8), (96, 8), contrabass_musicmaker_three],
2198     [(96, 8), (98, 8), contrabass_musicmaker_three],
2199     [(98, 8), (100, 8), contrabass_musicmaker_three],
2200     [(100, 8), (102, 8), contrabass_musicmaker_two],
2201     [(102, 8), (104, 8), contrabass_musicmaker_two],
2202 ]
2203 ])
2204
2205 all_timespan_lists = {
2206     'Voice 1': voice_1_timespan_list,
2207     'Voice 2': voice_2_timespan_list,
2208     'Voice 3': voice_3_timespan_list,
2209     'Voice 4': voice_4_timespan_list,
2210     'Voice 5': voice_5_timespan_list,
2211     'Voice 6': voice_6_timespan_list,
2212     'Voice 7': voice_7_timespan_list,
2213     'Voice 8': voice_8_timespan_list,
2214     'Voice 9': voice_9_timespan_list,
2215     'Voice 10': voice_10_timespan_list,
2216     'Voice 11': voice_11_timespan_list,
2217     'Voice 12': voice_12_timespan_list,
2218     'Voice 13': voice_13_timespan_list,
2219     'Voice 14': voice_14_timespan_list,
```

```

2220     'Voice 15': voice_15_timespan_list,
2221     'Voice 16': voice_16_timespan_list,
2222     'Voice 17': voice_17_timespan_list,
2223     'Voice 18': voice_18_timespan_list,
2224     'Voice 19': voice_19_timespan_list,
2225     'Voice 20': voice_20_timespan_list,
2226     'Voice 21': voice_21_timespan_list,
2227 }
2228
2229 global_timespan = abjad.Timespan(
2230     start_offset=0,
2231     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
2232 )
2233
2234 for voice_name, timespan_list in all_timespan_lists.items():
2235     silences = abjad.TimespanList([global_timespan])
2236     silences.extend(timespan_list)
2237     silences.sort()
2238     silences.compute_logical_xor()
2239     for silence_timespan in silences:
2240         timespan_list.append(
2241             abjad.AnnotatedTimespan(
2242                 start_offset=silence_timespan.start_offset,
2243                 stop_offset=silence_timespan.stop_offset,
2244                 annotation=MusicSpecifier(
2245                     music_maker=None,
2246                     voice_name=voice_name,
2247                 ),
2248             )
2249         )
2250     timespan_list.sort()
2251
2252 for voice_name, timespan_list in all_timespan_lists.items():
2253     shards = timespan_list.split_at_offsets(bounds)
2254     split_timespan_list = abjad.TimespanList()
2255     for shard in shards:
2256         split_timespan_list.extend(shard)
2257     split_timespan_list.sort()
2258     all_timespan_lists[voice_name] = timespan_list
2259
2260 score = abjad.Score([
2261     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
2262     abjad.StaffGroup(
2263         [
2264             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
2265             lilypond_type='Staff',),
2266             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
2267             lilypond_type='Staff',),
2268             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
2269             lilypond_type='Staff',),
2270             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
2271             lilypond_type='Staff',),
2272             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
2273             lilypond_type='Staff',),
2274             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
2275             lilypond_type='Staff',),
2276             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
2277             lilypond_type='Staff',),
2278             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',

```

```

2272     lilypond_type='Staff',),
2273         abjad.Staff([abjad.Voice(name='Voice 9')],name='Staff 9',
2274     lilypond_type='Staff',),
2275         abjad.Staff([abjad.Voice(name='Voice 10')],name='Staff 10',
2276     lilypond_type='Staff',),
2277         abjad.Staff([abjad.Voice(name='Voice 11')],name='Staff 11',
2278     lilypond_type='Staff',),
2279         abjad.Staff([abjad.Voice(name='Voice 12')],name='Staff 12',
2280     lilypond_type='Staff',),
2281         abjad.Staff([abjad.Voice(name='Voice 13')],name='Staff 13',
2282     lilypond_type='Staff',),
2283         abjad.Staff([abjad.Voice(name='Voice 14')],name='Staff 14',
2284     lilypond_type='Staff',),
2285         abjad.Staff([abjad.Voice(name='Voice 15')],name='Staff 15',
2286     lilypond_type='Staff',),
2287         abjad.Staff([abjad.Voice(name='Voice 16')],name='Staff 16',
2288     lilypond_type='Staff',),
2289         abjad.Staff([abjad.Voice(name='Voice 17')],name='Staff 17',
2290     lilypond_type='Staff',),
2291         abjad.Staff([abjad.Voice(name='Voice 18')],name='Staff 18',
2292     lilypond_type='Staff',),
2293         abjad.Staff([abjad.Voice(name='Voice 19')],name='Staff 19',
2294     lilypond_type='Staff',),
2295         abjad.Staff([abjad.Voice(name='Voice 20')],name='Staff 20',
2296     lilypond_type='Staff',),
2297         abjad.Staff([abjad.Voice(name='Voice 21')],name='Staff 21',
2298     lilypond_type='Staff',),
2299     ],
2300     name='Staff Group',
2301 )
2302 ])
2303
2304 for time_signature in time_signatures:
2305     skip = abjad.Skip(1, multiplier=(time_signature))
2306     abjad.attach(time_signature, skip)
2307     score['Global Context'].append(skip)
2308
2309 print('Making containers ...')
2310
2311 def make_container(music_maker, durations):
2312     selections = music_maker(durations)
2313     container = abjad.Container([])
2314     container.extend(selections)
2315     return container
2316
2317 def key_function(timespan):
2318     return timespan.annotation.music_maker or silence_maker
2319
2320 for voice_name, timespan_list in all_timespan_lists.items():
2321     for music_maker, grouper in itertools.groupby(
2322         timespan_list,
2323         key=key_function,
2324     ):
2325         durations = [timespan.duration for timespan in grouper]
2326         container = make_container(music_maker, durations)
2327         voice = score[voice_name]
2328         voice.append(container)
2329
2330 print('Splitting and rewriting ...')

```

```

2317 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
2318     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
2319         time_signature = time_signatures[i]
2320         abjad.mutate(shard).rewrite_meter(time_signature)
2321
2322 print('Beaming runs ...')
2323 for voice in abjad.select(score).components(abjad.Voice):
2324     for run in abjad.select(voice).runs():
2325         specifier = abjadext.rmakers.BeamSpecifier(
2326             beam_each_division=False,
2327         )
2328         specifier(run)
2329     abjad.beam(voice[:], beam_lone_notes=False, beam_rests=False, )
2330
2331 print('Stopping Hairpins ...')
2332 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2333     for rest in abjad.iterate(staff).components(abjad.Rest):
2334         previous_leaf = abjad.inspect(rest).leaf(-1)
2335         if isinstance(previous_leaf, abjad.Note):
2336             abjad.attach(abjad.StopHairpin(), rest)
2337         elif isinstance(previous_leaf, abjad.Chord):
2338             abjad.attach(abjad.StopHairpin(), rest)
2339         elif isinstance(previous_leaf, abjad.Rest):
2340             pass
2341
2342 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2343     first_leaf = abjad.select(staff).leaves()[0]
2344     stop = abjad.LilyPondLiteral(r'\!', format_slot='after', )
2345     abjad.attach(stop, first_leaf)
2346
2347 staffs = [staff for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff)]
2348
2349 scales = [
2350     soprano_random_walk_notes,
2351     soprano_1_random_walk_notes,
2352     soprano_2_random_walk_notes,
2353     soprano_3_random_walk_notes,
2354     alto_1_random_walk_notes,
2355     alto_2_random_walk_notes,
2356     alto_3_random_walk_notes,
2357     alto_4_random_walk_notes,
2358     alto_5_random_walk_notes,
2359     alto_6_random_walk_notes,
2360     tenor_1_random_walk_notes,
2361     tenor_2_random_walk_notes,
2362     tenor_3_random_walk_notes,
2363     tenor_4_random_walk_notes,
2364     tenor_5_random_walk_notes,
2365     baritone_1_random_walk_notes,
2366     baritone_2_random_walk_notes,
2367     baritone_3_random_walk_notes,
2368     bass_1_random_walk_notes,
2369     bass_2_random_walk_notes,
2370     contrabass_random_walk_notes,
2371 ]
2372
2373 for staff, scale in zip(staffs, scales):
2374     logical_ties = [i for i in abjad.iterate(staff).logical_ties(pitched=True)]

```

```

2375     pitches = cyc(scale)
2376     for i, logicl_tie in enumerate(logicl_ties):
2377         if logicl_tie.is_pitched ==True:
2378             pitch = next(pitches)
2379             for note in logicl_tie:
2380                 note.written_pitch = pitch
2381
2382 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2383     abjad.glissando(
2384         staff[:],
2385         allow_repeats=True,
2386     )
2387
2388 print('Adding attachments ...')
2389 bar_line = abjad.BarLine('||')
2390 metro = abjad.MetronomeMark((1, 4), 90)
2391 markup = abjad.Markup(r'\bold { B }')
2392 mark = abjad.RehearsalMark(markup=markup)
2393
2394 instruments = cyc([
2395     abjad.SopranoSaxophone(),
2396     abjad.SopranoSaxophone(),
2397     abjad.SopranoSaxophone(),
2398     abjad.SopranoSaxophone(),
2399     abjad.AltoSaxophone(),
2400     abjad.AltoSaxophone(),
2401     abjad.AltoSaxophone(),
2402     abjad.AltoSaxophone(),
2403     abjad.AltoSaxophone(),
2404     abjad.AltoSaxophone(),
2405     abjad.TenorSaxophone(),
2406     abjad.TenorSaxophone(),
2407     abjad.TenorSaxophone(),
2408     abjad.TenorSaxophone(),
2409     abjad.TenorSaxophone(),
2410     abjad.BaritoneSaxophone(),
2411     abjad.BaritoneSaxophone(),
2412     abjad.BaritoneSaxophone(),
2413     abjad.BassSaxophone(),
2414     abjad.BassSaxophone(),
2415     abjad.ContrabassSaxophone(),
2416 ])
2417
2418 abbreviations = cyc([
2419     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
2420     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
2421     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
2422     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
2423     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
2424     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
2425     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
2426     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
2427     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
2428     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
2429     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
2430     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
2431     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
2432     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
2433     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),

```

```

2434 abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
2435 abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
2436 abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
2437 abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
2438 abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
2439 abjad.MarginMarkup(markup=abjad.Markup('cbs.'),)
2440 ])
2441
2442 names = cyc([
2443     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
2444     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
2445     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
2446     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
2447     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
2448     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
2449     abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
2450     abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
2451     abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
2452     abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
2453     abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
2454     abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
2455     abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
2456     abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
2457     abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
2458     abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
2459     abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
2460     abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
2461     abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
2462     abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
2463 ])
2464 ]
2465
2466 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2467     leaf1 = abjad.select(staff).leaves()[0]
2468     abjad.attach(next(instruments), leaf1)
2469     abjad.attach(next(abbreviations), leaf1)
2470     abjad.attach(next(names), leaf1)
2471
2472 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
2473     leaf1 = abjad.select(staff).leaves()[0]
2474     last_leaf = abjad.select(staff).leaves()[-1]
2475     abjad.attach(metro, leaf1)
2476     abjad.attach(bar_line, last_leaf)
2477
2478 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
2479     leaf1 = abjad.select(staff).leaves()[0]
2480     abjad.attach(mark, leaf1)
2481
2482 score_file = abjad.LilyPondFile.new(
2483     score,
2484     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2485     _stylesheets/abjad.ily'],
2486 )
2487
2488 abjad.SegmentMaker.comment_measure_numbers(score)
2489 ######
2490 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_B'
2491 pdf_path = f'{directory}/Section_B.pdf'

```

```

2492 path = pathlib.Path('Section_B.pdf')
2493 if path.exists():
2494     print(f'Removing {pdf_path} ...')
2495     path.unlink()
2496 time_1 = time.time()
2497 print(f'Persisting {pdf_path} ...')
2498 result = abjad.persist(score_file).as_pdf(pdf_path)
2499 print(result[0])
2500 print(result[1])
2501 print(result[2])
2502 success = result[3]
2503 if success is False:
2504     print('LilyPond failed!')
2505 time_2 = time.time()
2506 total_time = time_2 - time_1
2507 print(f'Total time: {total_time} seconds')
2508 if path.exists():
2509     print(f'Opening {pdf_path} ...')
2510     os.system(f'open {pdf_path}')
2511 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_B/Section_B.ly').readlines()
2512 open('/Users/evansdsg2/Scores/guerrero/Build/Section_B.ly', 'w').writelines(score_lines[15:-1])
2513
2514 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
2515     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2516     signature_copy = abjad.mutate(signatures).copy()
2517     staff_copy = abjad.mutate(staff).copy()
2518     part = abjad.Score()
2519     part.insert(0, staff)
2520     part.insert(0, signature_copy)
2521     part_file = abjad.LilyPondFile.new(
2522         part,
2523         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/_stylesheets/abjad.ily'],
2524         )
2525     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano'
2526     pdf_path = f'{directory}/Section_B.pdf'
2527     path = pathlib.Path('Section_B.pdf')
2528     if path.exists():
2529         print(f'Removing {pdf_path} ...')
2530         path.unlink()
2531     time_1 = time.time()
2532     print(f'Persisting {pdf_path} ...')
2533     result = abjad.persist(part_file).as_pdf(pdf_path)
2534     print(result[0])
2535     print(result[1])
2536     print(result[2])
2537     success = result[3]
2538     if success is False:
2539         print('LilyPond failed!')
2540     time_2 = time.time()
2541     total_time = time_2 - time_1
2542     print(f'Total time: {total_time} seconds')
2543     if path.exists():
2544         print(f'Opening {pdf_path} ...')
2545         os.system(f'open {pdf_path}')
2546     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano'/Section_B.ly').readlines()

```

```

2547     open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano1/Section_B.ly',
2548         'w').writelines(part_lines[15:-1])
2549
2549 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
2550     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2551     signature_copy = abjad.mutate(signatures).copy()
2552     staff_copy = abjad.mutate(staff).copy()
2553     part = abjad.Score()
2554     part.insert(0, staff)
2555     part.insert(0, signature_copy)
2556     part_file = abjad.LilyPondFile.new(
2557         part,
2558         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2559 _stylesheets/abjad.ily'],
2560         )
2560     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1'
2561     pdf_path = f'{directory}/Section_B.pdf'
2562     path = pathlib.Path('Section_B.pdf')
2563     if path.exists():
2564         print(f'Removing {pdf_path} ...')
2565         path.unlink()
2566     time_1 = time.time()
2567     print(f'Persisting {pdf_path} ...')
2568     result = abjad.persist(part_file).as_pdf(pdf_path)
2569     print(result[0])
2570     print(result[1])
2571     print(result[2])
2572     success = result[3]
2573     if success is False:
2574         print('LilyPond failed!')
2575     time_2 = time.time()
2576     total_time = time_2 - time_1
2577     print(f'Total time: {total_time} seconds')
2578     if path.exists():
2579         print(f'Opening {pdf_path} ...')
2580         os.system(f'open {pdf_path}')
2581     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/
2582 Section_B.ly').readlines()
2583     open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/Section_B.ly',
2584         'w').writelines(part_lines[15:-1])
2585
2584 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
2585     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2586     signature_copy = abjad.mutate(signatures).copy()
2587     staff_copy = abjad.mutate(staff).copy()
2588     part = abjad.Score()
2589     part.insert(0, staff)
2590     part.insert(0, signature_copy)
2591     part_file = abjad.LilyPondFile.new(
2592         part,
2593         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2594 _stylesheets/abjad.ily'],
2595         )
2595     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2'
2596     pdf_path = f'{directory}/Section_B.pdf'
2597     path = pathlib.Path('Section_B.pdf')
2598     if path.exists():
2599         print(f'Removing {pdf_path} ...')
2600         path.unlink()

```

```

2601 time_1 = time.time()
2602 print(f'Persisting {pdf_path} ...')
2603 result = abjad.persist(part_file).as_pdf(pdf_path)
2604 print(result[0])
2605 print(result[1])
2606 print(result[2])
2607 success = result[3]
2608 if success is False:
2609     print('LilyPond failed!')
2610 time_2 = time.time()
2611 total_time = time_2 - time_1
2612 print(f'Total time: {total_time} seconds')
2613 if path.exists():
2614     print(f'Opening {pdf_path} ...')
2615     os.system(f'open {pdf_path}')
2616 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/
Section_B.ly').readlines()
2617 open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/Section_B.ly',
'w').writelines(part_lines[15:-1])
2618
2619 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
2620     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2621     signature_copy = abjad.mutate(signatures).copy()
2622     staff_copy = abjad.mutate(staff).copy()
2623     part = abjad.Score()
2624     part.insert(0, staff)
2625     part.insert(0, signature_copy)
2626     part_file = abjad.LilyPondFile.new(
2627         part,
2628         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2629         )
2630     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3'
2631     pdf_path = f'{directory}/Section_B.pdf'
2632     path = pathlib.Path('Section_B.pdf')
2633     if path.exists():
2634         print(f'Removing {pdf_path} ...')
2635         path.unlink()
2636     time_1 = time.time()
2637     print(f'Persisting {pdf_path} ...')
2638     result = abjad.persist(part_file).as_pdf(pdf_path)
2639     print(result[0])
2640     print(result[1])
2641     print(result[2])
2642     success = result[3]
2643     if success is False:
2644         print('LilyPond failed!')
2645     time_2 = time.time()
2646     total_time = time_2 - time_1
2647     print(f'Total time: {total_time} seconds')
2648     if path.exists():
2649         print(f'Opening {pdf_path} ...')
2650         os.system(f'open {pdf_path}')
2651     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/
Section_B.ly').readlines()
2652     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/Section_B.ly',
'w').writelines(part_lines[15:-1])
2653
2654 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):

```

```

2655 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2656 signature_copy = abjad.mutate(signatures).copy()
2657 staff_copy = abjad.mutate(staff).copy()
2658 part = abjad.Score()
2659 part.insert(0, staff)
2660 part.insert(0, signature_copy)
2661 part_file = abjad.LilyPondFile.new(
2662     part,
2663     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2664 _stylesheets/abjad.ily'],
2665     )
2666 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1'
2667 pdf_path = f'{directory}/Section_B.pdf'
2668 path = pathlib.Path('Section_B.pdf')
2669 if path.exists():
2670     print(f'Removing {pdf_path} ...')
2671     path.unlink()
2672 time_1 = time.time()
2673 print(f'Persisting {pdf_path} ...')
2674 result = abjad.persist(part_file).as_pdf(pdf_path)
2675 print(result[0])
2676 print(result[1])
2677 print(result[2])
2678 success = result[3]
2679 if success is False:
2680     print('LilyPond failed!')
2681 time_2 = time.time()
2682 total_time = time_2 - time_1
2683 print(f'Total time: {total_time} seconds')
2684 if path.exists():
2685     print(f'Opening {pdf_path} ...')
2686     os.system(f'open {pdf_path}')
2687 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/
2688 Section_B.ly').readlines()
2689 open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/Section_B.ly', 'w'
2690 .writelines(part_lines[15:-1])
2691
2692 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
2693     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2694     signature_copy = abjad.mutate(signatures).copy()
2695     staff_copy = abjad.mutate(staff).copy()
2696     part = abjad.Score()
2697     part.insert(0, staff)
2698     part.insert(0, signature_copy)
2699     part_file = abjad.LilyPondFile.new(
2700         part,
2701         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2702 _stylesheets/abjad.ily'],
2703         )
2704     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.) alto2'
2705     pdf_path = f'{directory}/Section_B.pdf'
2706     path = pathlib.Path('Section_B.pdf')
2707     if path.exists():
2708         print(f'Removing {pdf_path} ...')
2709         path.unlink()
2710     time_1 = time.time()
2711     print(f'Persisting {pdf_path} ...')
2712     result = abjad.persist(part_file).as_pdf(pdf_path)
2713     print(result[0])

```

```

2710     print(result[1])
2711     print(result[2])
2712     success = result[3]
2713     if success is False:
2714         print('LilyPond failed!')
2715     time_2 = time.time()
2716     total_time = time_2 - time_1
2717     print(f'Total time: {total_time} seconds')
2718     if path.exists():
2719         print(f'Opening {pdf_path} ...')
2720         os.system(f'open {pdf_path}')
2721     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
Section_B.ly').readlines()
2722     open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_B.ly', 'w'
).writelines(part_lines[15:-1])
2723
2724 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
2725     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2726     signature_copy = abjad.mutate(signatures).copy()
2727     staff_copy = abjad.mutate(staff).copy()
2728     part = abjad.Score()
2729     part.insert(0, staff)
2730     part.insert(0, signature_copy)
2731     part_file = abjad.LilyPondFile.new(
2732         part,
2733         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2734         )
2735     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
2736     pdf_path = f'{directory}/Section_B.pdf'
2737     path = pathlib.Path('Section_B.pdf')
2738     if path.exists():
2739         print(f'Removing {pdf_path} ...')
2740         path.unlink()
2741     time_1 = time.time()
2742     print(f'Persisting {pdf_path} ...')
2743     result = abjad.persist(part_file).as_pdf(pdf_path)
2744     print(result[0])
2745     print(result[1])
2746     print(result[2])
2747     success = result[3]
2748     if success is False:
2749         print('LilyPond failed!')
2750     time_2 = time.time()
2751     total_time = time_2 - time_1
2752     print(f'Total time: {total_time} seconds')
2753     if path.exists():
2754         print(f'Opening {pdf_path} ...')
2755         os.system(f'open {pdf_path}')
2756     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/
Section_B.ly').readlines()
2757     open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/Section_B.ly', 'w'
).writelines(part_lines[15:-1])
2758
2759 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
2760     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2761     signature_copy = abjad.mutate(signatures).copy()
2762     staff_copy = abjad.mutate(staff).copy()
2763     part = abjad.Score()

```

```

2764 part.insert(0, staff)
2765 part.insert(0, signature_copy)
2766 part_file = abjad.LilyPondFile.new(
2767     part,
2768     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2769 _stylesheets/abjad.ily'],
2770     )
2770 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
2771 pdf_path = f'{directory}/Section_B.pdf'
2772 path = pathlib.Path('Section_B.pdf')
2773 if path.exists():
2774     print(f'Removing {pdf_path} ...')
2775     path.unlink()
2776 time_1 = time.time()
2777 print(f'Persisting {pdf_path} ...')
2778 result = abjad.persist(part_file).as_pdf(pdf_path)
2779 print(result[0])
2780 print(result[1])
2781 print(result[2])
2782 success = result[3]
2783 if success is False:
2784     print('LilyPond failed!')
2785 time_2 = time.time()
2786 total_time = time_2 - time_1
2787 print(f'Total time: {total_time} seconds')
2788 if path.exists():
2789     print(f'Opening {pdf_path} ...')
2790     os.system(f'open {pdf_path}')
2791 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/
2792 Section_B.ly').readlines()
2793 open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/Section_B.ly', 'w'
2794 ).writelines(part_lines[15:-1])
2795
2796 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
2797     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2798     signature_copy = abjad.mutate(signatures).copy()
2799     staff_copy = abjad.mutate(staff).copy()
2800     part = abjad.Score()
2801     part.insert(0, staff)
2802     part.insert(0, signature_copy)
2803     part_file = abjad.LilyPondFile.new(
2804         part,
2805         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2806 _stylesheets/abjad.ily'],
2807         )
2808 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
2809 pdf_path = f'{directory}/Section_B.pdf'
2810 path = pathlib.Path('Section_B.pdf')
2811 if path.exists():
2812     print(f'Removing {pdf_path} ...')
2813     path.unlink()
2814 time_1 = time.time()
2815 print(f'Persisting {pdf_path} ...')
2816 result = abjad.persist(part_file).as_pdf(pdf_path)
2817 print(result[0])
2818 print(result[1])
2819 print(result[2])
2820 success = result[3]
2821 if success is False:

```

```

2819     print('LilyPond failed!')
2820 time_2 = time.time()
2821 total_time = time_2 - time_1
2822 print(f'Total time: {total_time} seconds')
2823 if path.exists():
2824     print(f'Opening {pdf_path} ...')
2825     os.system(f'open {pdf_path}')
2826 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/
Section_B.ly').readlines()
2827 open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/Section_B.ly', 'w'
).writelines(part_lines[15:-1])
2828
2829 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
2830     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2831     signature_copy = abjad.mutate(signatures).copy()
2832     staff_copy = abjad.mutate(staff).copy()
2833     part = abjad.Score()
2834     part.insert(0, staff)
2835     part.insert(0, signature_copy)
2836     part_file = abjad.LilyPondFile.new(
2837         part,
2838         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2839         )
2840     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
2841     pdf_path = f'{directory}/Section_B.pdf'
2842     path = pathlib.Path('Section_B.pdf')
2843     if path.exists():
2844         print(f'Removing {pdf_path} ...')
2845         path.unlink()
2846     time_1 = time.time()
2847     print(f'Persisting {pdf_path} ...')
2848     result = abjad.persist(part_file).as_pdf(pdf_path)
2849     print(result[0])
2850     print(result[1])
2851     print(result[2])
2852     success = result[3]
2853     if success is False:
2854         print('LilyPond failed!')
2855     time_2 = time.time()
2856     total_time = time_2 - time_1
2857     print(f'Total time: {total_time} seconds')
2858     if path.exists():
2859         print(f'Opening {pdf_path} ...')
2860         os.system(f'open {pdf_path}')
2861     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
Section_B.ly').readlines()
2862     open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_B.ly', 'w'
').writelines(part_lines[15:-1])
2863
2864 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
2865     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2866     signature_copy = abjad.mutate(signatures).copy()
2867     staff_copy = abjad.mutate(staff).copy()
2868     part = abjad.Score()
2869     part.insert(0, staff)
2870     part.insert(0, signature_copy)
2871     part_file = abjad.LilyPondFile.new(
2872         part,

```

```

2873     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2874 _stylesheets/abjad.ily'],
2875     )
2876 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
2877 pdf_path = f'{directory}/Section_B.pdf'
2878 path = pathlib.Path('Section_B.pdf')
2879 if path.exists():
2880     print(f'Removing {pdf_path} ...')
2881     path.unlink()
2882 time_1 = time.time()
2883 print(f'Persisting {pdf_path} ...')
2884 result = abjad.persist(part_file).as_pdf(pdf_path)
2885 print(result[0])
2886 print(result[1])
2887 print(result[2])
2888 success = result[3]
2889 if success is False:
2890     print('LilyPond failed!')
2891 time_2 = time.time()
2892 total_time = time_2 - time_1
2893 print(f'Total time: {total_time} seconds')
2894 if path.exists():
2895     print(f'Opening {pdf_path} ...')
2896     os.system(f'open {pdf_path}')
2897 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
2898 Section_B.ly').readlines()
2899 open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Section_B.ly', 'w').
2900 writelines(part_lines[15:-1])
2901
2902 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
2903     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2904     signature_copy = abjad.mutate(signatures).copy()
2905     staff_copy = abjad.mutate(staff).copy()
2906     part = abjad.Score()
2907     part.insert(0, staff)
2908     part.insert(0, signature_copy)
2909     part_file = abjad.LilyPondFile.new(
2910         part,
2911         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2912 _stylesheets/abjad.ily'],
2913         )
2914 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
2915 pdf_path = f'{directory}/Section_B.pdf'
2916 path = pathlib.Path('Section_B.pdf')
2917 if path.exists():
2918     print(f'Removing {pdf_path} ...')
2919     path.unlink()
2920 time_1 = time.time()
2921 print(f'Persisting {pdf_path} ...')
2922 result = abjad.persist(part_file).as_pdf(pdf_path)
2923 print(result[0])
2924 print(result[1])
2925 print(result[2])
2926 success = result[3]
2927 if success is False:
2928     print('LilyPond failed!')
2929 time_2 = time.time()
2930 total_time = time_2 - time_1
2931 print(f'Total time: {total_time} seconds')

```

```

2928     if path.exists():
2929         print(f'Opening {pdf_path} ...')
2930         os.system(f'open {pdf_path}')
2931     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/
2932     Section_B.ly').readlines()
2933     open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/Section_B.ly', ,
2934     w').writelines(part_lines[15:-1])
2935
2936 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
2937     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2938     signature_copy = abjad.mutate(signatures).copy()
2939     staff_copy = abjad.mutate(staff).copy()
2940     part = abjad.Score()
2941     part.insert(0, staff)
2942     part.insert(0, signature_copy)
2943     part_file = abjad.LilyPondFile.new(
2944         part,
2945         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2946         _stylesheets/abjad.ily'],
2947         )
2948     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3'
2949     pdf_path = f'{directory}/Section_B.pdf'
2950     path = pathlib.Path('Section_B.pdf')
2951     if path.exists():
2952         print(f'Removing {pdf_path} ...')
2953         path.unlink()
2954     time_1 = time.time()
2955     print(f'Persisting {pdf_path} ...')
2956     result = abjad.persist(part_file).as_pdf(pdf_path)
2957     print(result[0])
2958     print(result[1])
2959     print(result[2])
2960     success = result[3]
2961     if success is False:
2962         print('LilyPond failed!')
2963     time_2 = time.time()
2964     total_time = time_2 - time_1
2965     print(f'Total time: {total_time} seconds')
2966     if path.exists():
2967         print(f'Opening {pdf_path} ...')
2968         os.system(f'open {pdf_path}')
2969     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
2970     Section_B.ly').readlines()
2971     open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_B.ly', ,
2972     w').writelines(part_lines[15:-1])
2973
2974 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
2975     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2976     signature_copy = abjad.mutate(signatures).copy()
2977     staff_copy = abjad.mutate(staff).copy()
2978     part = abjad.Score()
2979     part.insert(0, staff)
2980     part.insert(0, signature_copy)
2981     part_file = abjad.LilyPondFile.new(
2982         part,
2983         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2984         _stylesheets/abjad.ily'],
2985         )
2986     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'

```

```

2981 pdf_path = f'{directory}/Section_B.pdf'
2982 path = pathlib.Path('Section_B.pdf')
2983 if path.exists():
2984     print(f'Removing {pdf_path} ...')
2985     path.unlink()
2986 time_1 = time.time()
2987 print(f'Persisting {pdf_path} ...')
2988 result = abjad.persist(part_file).as_pdf(pdf_path)
2989 print(result[0])
2990 print(result[1])
2991 print(result[2])
2992 success = result[3]
2993 if success is False:
2994     print('LilyPond failed!')
2995 time_2 = time.time()
2996 total_time = time_2 - time_1
2997 print(f'Total time: {total_time} seconds')
2998 if path.exists():
2999     print(f'Opening {pdf_path} ...')
3000     os.system(f'open {pdf_path}')
3001 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.)tenor4/
Section_B.ly').readlines()
3002 open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.)tenor4/Section_B.ly', ,
w').writelines(part_lines[15:-1])
3003
3004 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
3005     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3006     signature_copy = abjad.mutate(signatures).copy()
3007     staff_copy = abjad.mutate(staff).copy()
3008     part = abjad.Score()
3009     part.insert(0, staff)
3010     part.insert(0, signature_copy)
3011     part_file = abjad.LilyPondFile.new(
3012         part,
3013         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
3014         )
3015 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5'
3016 pdf_path = f'{directory}/Section_B.pdf'
3017 path = pathlib.Path('Section_B.pdf')
3018 if path.exists():
3019     print(f'Removing {pdf_path} ...')
3020     path.unlink()
3021 time_1 = time.time()
3022 print(f'Persisting {pdf_path} ...')
3023 result = abjad.persist(part_file).as_pdf(pdf_path)
3024 print(result[0])
3025 print(result[1])
3026 print(result[2])
3027 success = result[3]
3028 if success is False:
3029     print('LilyPond failed!')
3030 time_2 = time.time()
3031 total_time = time_2 - time_1
3032 print(f'Total time: {total_time} seconds')
3033 if path.exists():
3034     print(f'Opening {pdf_path} ...')
3035     os.system(f'open {pdf_path}')
3036 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/

```

```

3037     Section_B.ly').readlines()
3038     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/Section_B.ly', 'w').writelines(part_lines[15:-1])
3039
3040 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
3041     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3042     signature_copy = abjad.mutate(signatures).copy()
3043     staff_copy = abjad.mutate(staff).copy()
3044     part = abjad.Score()
3045     part.insert(0, staff)
3046     part.insert(0, signature_copy)
3047     part_file = abjad.LilyPondFile.new(
3048         part,
3049         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3050         _stylesheets/abjad.ily'],
3051         )
3052     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1'
3053     pdf_path = f'{directory}/Section_B.pdf'
3054     path = pathlib.Path('Section_B.pdf')
3055     if path.exists():
3056         print(f'Removing {pdf_path} ...')
3057         path.unlink()
3058     time_1 = time.time()
3059     print(f'Persisting {pdf_path} ...')
3060     result = abjad.persist(part_file).as_pdf(pdf_path)
3061     print(result[0])
3062     print(result[1])
3063     print(result[2])
3064     success = result[3]
3065     if success is False:
3066         print('LilyPond failed!')
3067     time_2 = time.time()
3068     total_time = time_2 - time_1
3069     print(f'Total time: {total_time} seconds')
3070     if path.exists():
3071         print(f'Opening {pdf_path} ...')
3072         os.system(f'open {pdf_path}')
3073     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/
3074     Section_B.ly').readlines()
3075     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/Section_B.ly',
3076     'w').writelines(part_lines[15:-1])
3077
3078 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
3079     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3080     signature_copy = abjad.mutate(signatures).copy()
3081     staff_copy = abjad.mutate(staff).copy()
3082     part = abjad.Score()
3083     part.insert(0, staff)
3084     part.insert(0, signature_copy)
3085     part_file = abjad.LilyPondFile.new(
3086         part,
3087         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3088         _stylesheets/abjad.ily'],
3089         )
3089     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2'
3090     pdf_path = f'{directory}/Section_B.pdf'
3091     path = pathlib.Path('Section_B.pdf')
3092     if path.exists():
3093         print(f'Removing {pdf_path} ...')

```

```

3090     path.unlink()
3091 time_1 = time.time()
3092 print(f'Persisting {pdf_path} ...')
3093 result = abjad.persist(part_file).as_pdf(pdf_path)
3094 print(result[0])
3095 print(result[1])
3096 print(result[2])
3097 success = result[3]
3098 if success is False:
3099     print('LilyPond failed!')
3100 time_2 = time.time()
3101 total_time = time_2 - time_1
3102 print(f'Total time: {total_time} seconds')
3103 if path.exists():
3104     print(f'Opening {pdf_path} ...')
3105     os.system(f'open {pdf_path}')
3106 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/
Section_B.ly').readlines()
3107 open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/Section_B.ly',
      'w').writelines(part_lines[15:-1])
3108
3109 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
3110 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3111 signature_copy = abjad.mutate(signatures).copy()
3112 staff_copy = abjad.mutate(staff).copy()
3113 part = abjad.Score()
3114 part.insert(0, staff)
3115 part.insert(0, signature_copy)
3116 part_file = abjad.LilyPondFile.new(
3117     part,
3118     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
3119     )
3120 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3'
3121 pdf_path = f'{directory}/Section_B.pdf'
3122 path = pathlib.Path('Section_B.pdf')
3123 if path.exists():
3124     print(f'Removing {pdf_path} ...')
3125     path.unlink()
3126 time_1 = time.time()
3127 print(f'Persisting {pdf_path} ...')
3128 result = abjad.persist(part_file).as_pdf(pdf_path)
3129 print(result[0])
3130 print(result[1])
3131 print(result[2])
3132 success = result[3]
3133 if success is False:
3134     print('LilyPond failed!')
3135 time_2 = time.time()
3136 total_time = time_2 - time_1
3137 print(f'Total time: {total_time} seconds')
3138 if path.exists():
3139     print(f'Opening {pdf_path} ...')
3140     os.system(f'open {pdf_path}')
3141 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/
Section_B.ly').readlines()
3142 open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/Section_B.ly',
      'w').writelines(part_lines[15:-1])
3143

```

```

3144 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
3145     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3146     signature_copy = abjad.mutate(signatures).copy()
3147     staff_copy = abjad.mutate(staff).copy()
3148     part = abjad.Score()
3149     part.insert(0, staff)
3150     part.insert(0, signature_copy)
3151     part_file = abjad.LilyPondFile.new(
3152         part,
3153         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
3154         _stylesheets/abjad.ly'],
3155         )
3156     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
3157     pdf_path = f'{directory}/Section_B.pdf'
3158     path = pathlib.Path('Section_B.pdf')
3159     if path.exists():
3160         print(f'Removing {pdf_path} ...')
3161         path.unlink()
3162     time_1 = time.time()
3163     print(f'Persisting {pdf_path} ...')
3164     result = abjad.persist(part_file).as_pdf(pdf_path)
3165     print(result[0])
3166     print(result[1])
3167     print(result[2])
3168     success = result[3]
3169     if success is False:
3170         print('LilyPond failed!')
3171     time_2 = time.time()
3172     total_time = time_2 - time_1
3173     print(f'Total time: {total_time} seconds')
3174     if path.exists():
3175         print(f'Opening {pdf_path} ...')
3176         os.system(f'open {pdf_path}')
3177     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
3178     Section_B.ly').readlines()
3179     open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_B.ly', 'w'
3180     .writelines(part_lines[15:-1])
3181
3182 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
3183     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3184     signature_copy = abjad.mutate(signatures).copy()
3185     staff_copy = abjad.mutate(staff).copy()
3186     part = abjad.Score()
3187     part.insert(0, staff)
3188     part.insert(0, signature_copy)
3189     part_file = abjad.LilyPondFile.new(
3190         part,
3191         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
3192         _stylesheets/abjad.ly'],
3193         )
3194     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
3195     pdf_path = f'{directory}/Section_B.pdf'
3196     path = pathlib.Path('Section_B.pdf')
3197     if path.exists():
3198         print(f'Removing {pdf_path} ...')

```

```

3199     print(result[0])
3200     print(result[1])
3201     print(result[2])
3202     success = result[3]
3203     if success is False:
3204         print('LilyPond failed!')
3205     time_2 = time.time()
3206     total_time = time_2 - time_1
3207     print(f'Total time: {total_time} seconds')
3208     if path.exists():
3209         print(f'Opening {pdf_path} ...')
3210         os.system(f'open {pdf_path}')
3211     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
Section_B.ly').readlines()
3212     open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_B.ly', 'w'
').writelines(part_lines[15:-1])
3213
3214 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
3215     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3216     signature_copy = abjad.mutate(signatures).copy()
3217     staff_copy = abjad.mutate(staff).copy()
3218     part = abjad.Score()
3219     part.insert(0, staff)
3220     part.insert(0, signature_copy)
3221     part_file = abjad.LilyPondFile.new(
3222         part,
3223         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
3224         )
3225     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
3226     pdf_path = f'{directory}/Section_B.pdf'
3227     path = pathlib.Path('Section_B.pdf')
3228     if path.exists():
3229         print(f'Removing {pdf_path} ...')
3230         path.unlink()
3231     time_1 = time.time()
3232     print(f'Persisting {pdf_path} ...')
3233     result = abjad.persist(part_file).as_pdf(pdf_path)
3234     print(result[0])
3235     print(result[1])
3236     print(result[2])
3237     success = result[3]
3238     if success is False:
3239         print('LilyPond failed!')
3240     time_2 = time.time()
3241     total_time = time_2 - time_1
3242     print(f'Total time: {total_time} seconds')
3243     if path.exists():
3244         print(f'Opening {pdf_path} ...')
3245         os.system(f'open {pdf_path}')
3246     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/
Section_B.ly').readlines()
3247     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_B.ly
', 'w').writelines(part_lines[15:-1])

```

Listing 3.3: Invocation Source Code

### 3.4 Section C

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11
12 print('Interpreting file ...')
13
14 time_signatures = [
15     abjad.TimeSignature(pair) for pair in [
16         (4, 4), (4, 4), (4, 4), (4, 4),
17     ]
18 ]
19
20 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
21
22 def cyc(lst):
23     count = 0
24     while True:
25         yield lst[count%len(lst)]
26         count += 1
27
28 def grouper(lst1, lst2):
29     def cyc(lst):
30         c = 0
31         while True:
32             yield lst[c%len(lst)]
33             c += 1
34     lst1 = cyc(lst1)
35     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2]
36
37 def reduceMod(list_length, rw):
38     return [(x % list_length) for x in rw]
39
40
41 soprano_note = [27, 17, 8, 0, 11, ]
42 soprano_1_note = [[13.25, 16, 26.25, ], ]
43 soprano_2_note = [[13, 14.75, 26.25, ], ]
44 soprano_3_note = [[12.75, 15.5, 26, ], ]
45 alto_1_note = [[12.5, 19, 27.75, 34, ], ]
46 alto_2_note = [[12.5, 15.25, 25.5, ], ]
47 alto_3_note = [[1.75, 13.5, 22.25, 27, 30, ], ]
48 alto_4_note = [[12.5, 15.25, 25.5, ], ]
49 alto_5_note = [[1.75, 13.5, 22.25, 27, 30, ], ]
50 alto_6_note = [[12.5, 19, 27.75, 34, ], ]
51 tenor_1_note = [[6, 17.5, ], ]
52 tenor_2_note = [[6, 17.5, 25.5, 30, ], ]
53 tenor_3_note = [[6, 17.5, 25.5, 30.75, ], ]
54 tenor_4_note = [[6, 17.5, ], ]
55 tenor_5_note = [[6, 17.5, 25.5, 30.75, ], ]

```

```

56 baritone_1_note = [[13.25, 27.5, 33.75, ], ]
57 baritone_2_note = [[4, 16.5, 23.5, ], ]
58 baritone_3_note = [[7.75, 17.75, 25.5, 34, ], ]
59 bass_1_note = [9, 11, 11, 9, 11, ]
60 bass_2_note = [11, 9, 9, 11, 9, ]
61 contrabass_note = [-2, 2, 7, -2, 2, 7, 2, -2, ]
62
63
64 # -3 at bottom of chord for completion
65 sopranino_chord = [17, 27, 11, 0, 8,]
66 soprano_1_chord = [22, [13.25, 16, 26.25, ], 16, [13.25, 16, 26.25, ], [13.25, 16,
   26.25, ], ]
67 soprano_2_chord = [16, [13, 14.75, 26.25, ], 13, [13, 14.75, 26.25, ], 14, [13,
   14.75, 26.25, ], 13, [13, 14.75, 26.25, ], ]
68 soprano_3_chord = [13, [12.75, 15.5, 26, ], 13, 12, [12.75, 15.5, 26, ], 16,
   [12.75, 15.5, 26, ], 12, ]
69 alto_1_chord = [20, [12.5, 19, 27.75, 34, ], 12, [12.5, 19, 27.75, 34, ], 23, ]
70 alto_2_chord = [12, [12.5, 15.25, 25.5, ], 1, [12.5, 15.25, 25.5, ], 12, [12.5,
   15.25, 25.5, ], 23, ]
71 alto_3_chord = [1, [1.75, 13.5, 22.25, 27, 30, ], 23, [1.75, 13.5, 22.25, 27, 30,
   ], 1, [1.75, 13.5, 22.25, 27, 30, ], 12, [1.75, 13.5, 22.25, 27, 30, ], ]
72 alto_4_chord = [20, [12.5, 15.25, 25.5, ], 12, [12.5, 15.25, 25.5, ], 20, [12.5,
   15.25, 25.5, ], 23, ]
73 alto_5_chord = [12, [1.75, 13.5, 22.25, 27, 30, ], 20, 1, [1.75, 13.5, 22.25, 27,
   30, ], 23, [1.75, 13.5, 22.25, 27, 30, ], ]
74 alto_6_chord = [1, [12.5, 19, 27.75, 34, ], 20, 12, [12.5, 19, 27.75, 34, ], ]
75 tenor_1_chord = [17, [6, 17.5, ], -1, 25, [6, 17.5, ], 6, [6, 17.5, ], ]
76 tenor_2_chord = [6, [6, 17.5, 25.5, 30, ], 25, [6, 17.5, 25.5, 30, ], 17, ]
77 tenor_3_chord = [-1, [6, 17.5, 25.5, 30.75, ], 6, [6, 17.5, 25.5, 30.75, ], 17,
   [6, 17.5, 25.5, 30.75, ], 25, [6, 17.5, 25.5, 30.75, ], ]
78 tenor_4_chord = [17, [6, 17.5, ], -1, [6, 17.5, ], 6, [6, 17.5, ], ]
79 tenor_5_chord = [6, [6, 17.5, 25.5, 30.75, ], 25, [6, 17.5, 25.5, 30.75, ], 17,
   [6, 17.5, 25.5, 30.75, ], ]
80 baritone_1_chord = [13, [13.25, 27.5, 33.75, ], 24, [13.25, 27.5, 33.75, ], 13,
   [13.25, 27.5, 33.75, ], 6, [13.25, 27.5, 33.75, ], ]
81 baritone_2_chord = [6, [4, 16.5, 23.5, ], 24, 13, [4, 16.5, 23.5, ], 4, ]
82 baritone_3_chord = [4, [7.75, 17.75, 25.5, 34, ], 13, [7.75, 17.75, 25.5, 34, ],
   6, [7.75, 17.75, 25.5, 34, ], 24, [7.75, 17.75, 25.5, 34, ], ]
83 bass_1_chord = [11, 18, 9, 11, 0, ]
84 bass_2_chord = [9, 11, 0, 11, 9, 18, ]
85 contrabass_chord = [-2, 7, 2, 16, 7, 18, 16, 25, 16, ]
86
87 def reduceMod(x, rw):
88     return [(y % x) for y in rw]
89
90 seed(22)
91 sopranino_random_walk = []
92 sopranino_random_walk.append(-1 if random() < 0.5 else 1)
93 for i in range(1, 1000):
94     movement = -1 if random() < 0.5 else 1
95     value = sopranino_random_walk[i-1] + movement
96     sopranino_random_walk.append(value)
97     sopranino_walk_chord = [18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14,
   13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5, 6,
   7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ]
98 l = len(sopranino_walk_chord)
99 sopranino_random_walk_notes = [sopranino_walk_chord[x] for x in reduceMod(l,
   sopranino_random_walk)]
100

```

```

101 seed(23)
102 soprano_1_random_walk = []
103 soprano_1_random_walk.append(-1 if random() < 0.5 else 1)
104 for i in range(1, 1000):
105     movement = -1 if random() < 0.5 else 1
106     value = soprano_1_random_walk[i-1] + movement
107     soprano_1_random_walk.append(value)
108 soprano_1_walk_chord = [17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14,
109    13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5, 6,
110    7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ]
111 l = len(soprano_1_walk_chord)
112 soprano_1_random_walk_notes = [soprano_1_walk_chord[x] for x in reduceMod(l,
113    soprano_1_random_walk)]
114
115 seed(24)
116 soprano_2_random_walk = []
117 soprano_2_random_walk.append(-1 if random() < 0.5 else 1)
118 for i in range(1, 1000):
119     movement = -1 if random() < 0.5 else 1
120     value = soprano_2_random_walk[i-1] + movement
121     soprano_2_random_walk.append(value)
122 soprano_2_random_walk.append(value)
123 soprano_2_walk_chord = [16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15,
124    14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5,
125    6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ]
126 l = len(soprano_2_walk_chord)
127 soprano_2_random_walk_notes = [soprano_2_walk_chord[x] for x in reduceMod(l,
128    soprano_2_random_walk)]
129
130 seed(25)
131 soprano_3_random_walk = []
132 soprano_3_random_walk.append(-1 if random() < 0.5 else 1)
133 for i in range(1, 1000):
134     movement = -1 if random() < 0.5 else 1
135     value = soprano_3_random_walk[i-1] + movement
136     soprano_3_random_walk.append(value)
137 soprano_3_random_walk.append(value)
138 soprano_3_walk_chord = [15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16,
139    15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3,
140    4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ]
141 l = len(soprano_3_walk_chord)
142 soprano_3_random_walk_notes = [soprano_3_walk_chord[x] for x in reduceMod(l,
143    soprano_3_random_walk)]
144
145 seed(26)
146 alto_1_random_walk = []
147 alto_1_random_walk.append(-1 if random() < 0.5 else 1)
148 for i in range(1, 1000):
149     movement = -1 if random() < 0.5 else 1
150     value = alto_1_random_walk[i-1] + movement
151     alto_1_random_walk.append(value)
152 alto_1_random_walk.append(value)
153 alto_1_walk_chord = [14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16,
154    15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3,
155    4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ]
156 l = len(alto_1_walk_chord)
157 alto_1_random_walk_notes = [alto_1_walk_chord[x] for x in reduceMod(l,
158    alto_1_random_walk)]
159
160 seed(27)

```

```

148 alto_2_random_walk = []
149 alto_2_random_walk.append(-1 if random() < 0.5 else 1)
150 for i in range(1, 1000):
151     movement = -1 if random() < 0.5 else 1
152     value = alto_2_random_walk[i-1] + movement
153     alto_2_random_walk.append(value)
154 alto_2_walk_chord = [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17,
155     16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2,
156     3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ]
157 l = len(alto_2_walk_chord)
158 alto_2_random_walk_notes = [alto_2_walk_chord[x] for x in reduceMod(l,
159     alto_2_random_walk)]
160
161 seed(28)
162 alto_3_random_walk = []
163 alto_3_random_walk.append(-1 if random() < 0.5 else 1)
164 for i in range(1, 1000):
165     movement = -1 if random() < 0.5 else 1
166     value = alto_3_random_walk[i-1] + movement
167     alto_3_random_walk.append(value)
168 alto_3_walk_chord = [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18,
169     17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1,
170     2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ]
171 l = len(alto_3_walk_chord)
172 alto_3_random_walk_notes = [alto_3_walk_chord[x] for x in reduceMod(l,
173     alto_3_random_walk)]
174
175 seed(29)
176 alto_4_random_walk = []
177 alto_4_random_walk.append(-1 if random() < 0.5 else 1)
178 for i in range(1, 1000):
179     movement = -1 if random() < 0.5 else 1
180     value = alto_4_random_walk[i-1] + movement
181     alto_4_random_walk.append(value)
182 alto_4_walk_chord = [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19,
183     18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1,
184     0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ]
185 l = len(alto_4_walk_chord)
186 alto_4_random_walk_notes = [alto_4_walk_chord[x] for x in reduceMod(l,
187     alto_4_random_walk)]
188
189 seed(30)
190 alto_5_random_walk = []
191 alto_5_random_walk.append(-1 if random() < 0.5 else 1)
192 for i in range(1, 1000):
193     movement = -1 if random() < 0.5 else 1
194     value = alto_5_random_walk[i-1] + movement
195     alto_5_random_walk.append(value)
196 alto_5_walk_chord = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20,
197     19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2,
198     -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ]
199 l = len(alto_5_walk_chord)
200 alto_5_random_walk_notes = [alto_5_walk_chord[x] for x in reduceMod(l,
201     alto_5_random_walk)]
202
203 seed(31)
204 alto_6_random_walk = []
205 alto_6_random_walk.append(-1 if random() < 0.5 else 1)
206 for i in range(1, 1000):

```

```

195     movement = -1 if random() < 0.5 else 1
196     value = alto_6_random_walk[i-1] + movement
197     alto_6_random_walk.append(value)
198 alto_6_walk_chord = [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21,
199   20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1,
200   -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, ]
201 l = len(alto_6_walk_chord)
202 alto_6_random_walk_notes = [alto_6_walk_chord[x] for x in reduceMod(l,
203   alto_6_random_walk)]
204
205 seed(32)
206 tenor_1_random_walk = []
207 tenor_1_random_walk.append(-1 if random() < 0.5 else 1)
208 for i in range(1, 1000):
209     movement = -1 if random() < 0.5 else 1
210     value = tenor_1_random_walk[i-1] + movement
211     tenor_1_random_walk.append(value)
212 tenor_1_walk_chord = [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
213   21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
214   -1, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, ]
215 l = len(tenor_1_walk_chord)
216 tenor_1_random_walk_notes = [tenor_1_walk_chord[x] for x in reduceMod(l,
217   tenor_1_random_walk)]
218
219 seed(33)
220 tenor_2_random_walk = []
221 tenor_2_random_walk.append(-1 if random() < 0.5 else 1)
222 for i in range(1, 1000):
223     movement = -1 if random() < 0.5 else 1
224     value = tenor_2_random_walk[i-1] + movement
225     tenor_2_random_walk.append(value)
226 tenor_2_walk_chord = [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
227   21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
228   -1, -2, -1, 0, 1, 2, 3, 4, 5, 6, ]
229 l = len(tenor_2_walk_chord)
230 tenor_2_random_walk_notes = [tenor_2_walk_chord[x] for x in reduceMod(l,
231   tenor_2_random_walk)]
232
233 seed(34)
234 tenor_3_random_walk = []
235 tenor_3_random_walk.append(-1 if random() < 0.5 else 1)
236 for i in range(1, 1000):
237     movement = -1 if random() < 0.5 else 1
238     value = tenor_3_random_walk[i-1] + movement
239     tenor_3_random_walk.append(value)
240 tenor_3_walk_chord = [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
241   22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1,
242   0, -1, -2, -1, 0, 1, 2, 3, 4, 5, ]
243 l = len(tenor_3_walk_chord)
244 tenor_3_random_walk_notes = [tenor_3_walk_chord[x] for x in reduceMod(l,
245   tenor_3_random_walk)]
246
247 seed(35)
248 tenor_4_random_walk = []
249 tenor_4_random_walk.append(-1 if random() < 0.5 else 1)
250 for i in range(1, 1000):
251     movement = -1 if random() < 0.5 else 1
252     value = tenor_4_random_walk[i-1] + movement
253     tenor_4_random_walk.append(value)

```

```

242 tenor_4_walk_chord = [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
243   21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2,
244   1, 0, -1, -2, -1, 0, 1, 2, 3, 4, ]
245 l = len(tenor_4_walk_chord)
246 tenor_4_random_walk_notes = [tenor_4_walk_chord[x] for x in reduceMod(l,
247   tenor_4_random_walk)]
248
249 seed(36)
250 tenor_5_random_walk = []
251 tenor_5_random_walk.append(-1 if random() < 0.5 else 1)
252 for i in range(1, 1000):
253   movement = -1 if random() < 0.5 else 1
254   value = tenor_5_random_walk[i-1] + movement
255   tenor_5_random_walk.append(value)
256 tenor_5_walk_chord = [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
257   20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4,
258   3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, ]
259 l = len(tenor_5_walk_chord)
260 tenor_5_random_walk_notes = [tenor_5_walk_chord[x] for x in reduceMod(l,
261   tenor_5_random_walk)]
262
263 seed(37)
264 baritone_1_random_walk = []
265 baritone_1_random_walk.append(-1 if random() < 0.5 else 1)
266 for i in range(1, 1000):
267   movement = -1 if random() < 0.5 else 1
268   value = baritone_1_random_walk[i-1] + movement
269   baritone_1_random_walk.append(value)
270 baritone_1_walk_chord = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
271   19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5,
272   4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, ]
273 l = len(baritone_1_walk_chord)
274 baritone_1_random_walk_notes = [baritone_1_walk_chord[x] for x in reduceMod(l,
275   baritone_1_random_walk)]
276
277 seed(38)
278 baritone_2_random_walk = []
279 baritone_2_random_walk.append(-1 if random() < 0.5 else 1)
280 for i in range(1, 1000):
281   movement = -1 if random() < 0.5 else 1
282   value = baritone_2_random_walk[i-1] + movement
283   baritone_2_random_walk.append(value)
284 baritone_2_walk_chord = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
285   18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,
286   5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, ]
287 l = len(baritone_2_walk_chord)
288 baritone_2_random_walk_notes = [baritone_2_walk_chord[x] for x in reduceMod(l,
289   baritone_2_random_walk)]
290
291 seed(39)
292 baritone_3_random_walk = []
293 baritone_3_random_walk.append(-1 if random() < 0.5 else 1)
294 for i in range(1, 1000):
295   movement = -1 if random() < 0.5 else 1
296   value = baritone_3_random_walk[i-1] + movement
297   baritone_3_random_walk.append(value)
298 baritone_3_walk_chord = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
299   17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8,
300   7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, ]
301

```

```

287 l = len(baritone_3_walk_chord)
288 baritone_3_random_walk_notes = [baritone_3_walk_chord[x] for x in reduceMod(l,
289     baritone_3_random_walk)]
290
290 seed(40)
291 bass_1_random_walk = []
292 bass_1_random_walk.append(-1 if random() < 0.5 else 1)
293 for i in range(1, 1000):
294     movement = -1 if random() < 0.5 else 1
295     value = bass_1_random_walk[i-1] + movement
296     bass_1_random_walk.append(value)
297 bass_1_walk_chord = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
298     18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7,
299     6, 5, 4, 3, 2, 1, 0, -1, -2, -1, ]
300 l = len(bass_1_walk_chord)
301 bass_1_random_walk_notes = [bass_1_walk_chord[x] for x in reduceMod(l,
302     bass_1_random_walk)]
303
301 seed(41)
302 bass_2_random_walk = []
303 bass_2_random_walk.append(-1 if random() < 0.5 else 1)
304 for i in range(1, 1000):
305     movement = -1 if random() < 0.5 else 1
306     value = bass_2_random_walk[i-1] + movement
307     bass_2_random_walk.append(value)
308 bass_2_walk_chord = [-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
309     17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8,
310     7, 6, 5, 4, 3, 2, 1, 0, -1, -2, ]
311 l = len(bass_2_walk_chord)
310 bass_2_random_walk_notes = [bass_2_walk_chord[x] for x in reduceMod(l,
311     bass_2_random_walk)]
312
312 seed(42)
313 contrabass_random_walk = []
314 contrabass_random_walk.append(-1 if random() < 0.5 else 1)
315 for i in range(1, 1000):
316     movement = -1 if random() < 0.5 else 1
317     value = contrabass_random_walk[i-1] + movement
318     contrabass_random_walk.append(value)
319 contrabass_walk_chord = [-2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
320     15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11,
321     10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, ]
322 l = len(contrabass_walk_chord)
321 contrabass_random_walk_notes = [contrabass_walk_chord[x] for x in reduceMod(l,
322     contrabass_random_walk)]
323
323 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
324     talea=abjadext.rmakers.Talea(
325         counts=[3, 2, 1, 2, 3, 1, 3, 2, 3, 1, 2, ],
326         denominator=16,
327         ),
328     beamSpecifier=abjadext.rmakers.BeamSpecifier(
329         beamDivisionsTogether=True,
330         beamRests=False,
331         ),
332     extraCountsPerDivision=[1, 0, -1, 1, -1, 0, 1, ],
333     logicalTieMasks=[
334         abjadext.rmakers.silence([5], 7),
335         ],

```

```

336     division_masks=[  
337         abjadext.rmakers.SilenceMask(  
338             pattern=abjad.index([4], 9),  
339         ),  
340     ],  
341     tupletSpecifier=abjadext.rmakers.TupletSpecifier(  
342         trivialize=True,  
343         extract_trivial=True,  
344         rewrite_rest_filled=True,  
345         rewrite_sustained=True,  
346     ),  
347 )  
348  
349 rmaker_two = abjadext.rmakers.TaleaRhythmMaker(  
350     talea=abjadext.rmakers.Talea(  
351         counts=[1, 1, 1, 1, -1, 2, 2, 1, 2, 1, 3, -2, 2, 3, 2, 1, 1, 2, -1, 1, 1,  
352         3, ],  
353         denominator=16,  
354     ),  
355     beamSpecifier=abjadext.rmakers.BeamSpecifier(  
356         beam_divisions_together=True,  
357         beam_rests=False,  
358     ),  
359     extra_counts_per_division=[-1, 0, 1, -1, 1, 0, ],  
360     tupletSpecifier=abjadext.rmakers.TupletSpecifier(  
361         trivialize=True,  
362         extract_trivial=True,  
363         rewrite_rest_filled=True,  
364         rewrite_sustained=True,  
365     ),  
366 )  
367  
368 rmaker_three = abjadext.rmakers.NoteRhythmMaker()  
369  
370 attachment_handler_one = AttachmentHandler(  
371     starting_dynamic='mf',  
372     # ending_dynamic='f',  
373     hairpin='>o',  
374     articulation_list=['accent', '', 'tenuto', '', '', 'portato', '', '', '', ],  
375 )  
376 attachment_handler_two = AttachmentHandler(  
377     starting_dynamic='f',  
378     ending_dynamic='p',  
379     hairpin='>',  
380     articulation_list=['flageolet', 'flageolet', 'flageolet', 'flageolet', '  
381     flageolet', 'halfopen', 'halfopen', 'halfopen', '', '', '', 'halfopen', '  
382     flageolet', '', '', '', 'stopped', 'stopped', 'stopped', 'stopped', '', '  
383     halfopen', 'flageolet', 'halfopen', 'halfopen', 'halfopen', '', '', '', '  
384     stopped', 'stopped', 'stopped', 'stopped', '', '', '', ''],  
385 )  
386 attachment_handler_three = AttachmentHandler(  
387     starting_dynamic='mp',  
388     ending_dynamic='pp',  
389     hairpin='--',  
390     articulation_list=['espressivo'],  
391 )

```

```
390 #####sopranino#####
391 sopranino_musicmaker_one = MusicMaker(
392     rmaker=rmaker_one,
393     pitches=sopranino_chord,
394     continuous=True,
395     attachment_handler=attachment_handler_one,
396 )
397 sopranino_musicmaker_two = MusicMaker(
398     rmaker=rmaker_two,
399     pitches=sopranino_random_walk_notes,
400     continuous=True,
401     attachment_handler=attachment_handler_two,
402 )
403 sopranino_musicmaker_three = MusicMaker(
404     rmaker=rmaker_three,
405     pitches=sopranino_note,
406     continuous=True,
407     attachment_handler=attachment_handler_three,
408 )
409 #####soprano_one#####
410 soprano_one_musicmaker_one = MusicMaker(
411     rmaker=rmaker_one,
412     pitches=soprano_1_chord,
413     continuous=True,
414     attachment_handler=attachment_handler_one,
415 )
416 soprano_one_musicmaker_two = MusicMaker(
417     rmaker=rmaker_two,
418     pitches=soprano_1_random_walk_notes,
419     continuous=True,
420     attachment_handler=attachment_handler_two,
421 )
422 soprano_one_musicmaker_three = MusicMaker(
423     rmaker=rmaker_three,
424     pitches=soprano_1_note,
425     continuous=True,
426     attachment_handler=attachment_handler_three,
427 )
428 #####soprano_two#####
429 soprano_two_musicmaker_one = MusicMaker(
430     rmaker=rmaker_one,
431     pitches=soprano_2_chord,
432     continuous=True,
433     attachment_handler=attachment_handler_one,
434 )
435 soprano_two_musicmaker_two = MusicMaker(
436     rmaker=rmaker_two,
437     pitches=soprano_2_random_walk_notes,
438     continuous=True,
439     attachment_handler=attachment_handler_two,
440 )
441 soprano_two_musicmaker_three = MusicMaker(
442     rmaker=rmaker_three,
443     pitches=soprano_2_note,
444     continuous=True,
445     attachment_handler=attachment_handler_three,
446 )
447 #####soprano_three#####
448 soprano_three_musicmaker_one = MusicMaker(
```

```
449     rmaker=rmaker_one,
450     pitches=soprano_3_chord,
451     continuous=True,
452     attachment_handler=attachment_handler_one,
453 )
454 soprano_three_musicmaker_two = MusicMaker(
455     rmaker=rmaker_two,
456     pitches=soprano_3_random_walk_notes,
457     continuous=True,
458     attachment_handler=attachment_handler_two,
459 )
460 soprano_three_musicmaker_three = MusicMaker(
461     rmaker=rmaker_three,
462     pitches=soprano_3_note,
463     continuous=True,
464     attachment_handler=attachment_handler_three,
465 )
466 #####alto_one#####
467 alto_one_musicmaker_one = MusicMaker(
468     rmaker=rmaker_one,
469     pitches=alto_1_chord,
470     continuous=True,
471     attachment_handler=attachment_handler_one,
472 )
473 alto_one_musicmaker_two = MusicMaker(
474     rmaker=rmaker_two,
475     pitches=alto_1_random_walk_notes,
476     continuous=True,
477     attachment_handler=attachment_handler_two,
478 )
479 alto_one_musicmaker_three = MusicMaker(
480     rmaker=rmaker_three,
481     pitches=alto_1_note,
482     continuous=True,
483     attachment_handler=attachment_handler_three,
484 )
485 #####alto_two#####
486 alto_two_musicmaker_one = MusicMaker(
487     rmaker=rmaker_one,
488     pitches=alto_2_chord,
489     continuous=True,
490     attachment_handler=attachment_handler_one,
491 )
492 alto_two_musicmaker_two = MusicMaker(
493     rmaker=rmaker_two,
494     pitches=alto_2_random_walk_notes,
495     continuous=True,
496     attachment_handler=attachment_handler_two,
497 )
498 alto_two_musicmaker_three = MusicMaker(
499     rmaker=rmaker_three,
500     pitches=alto_2_note,
501     continuous=True,
502     attachment_handler=attachment_handler_three,
503 )
504 #####alto_three#####
505 alto_three_musicmaker_one = MusicMaker(
506     rmaker=rmaker_one,
507     pitches=alto_3_chord,
```

```
508     continuous=True,
509     attachment_handler=attachment_handler_one,
510 )
511 alto_three_musicmaker_two = MusicMaker(
512     rmaker=rmaker_two,
513     pitches=alto_3_random_walk_notes,
514     continuous=True,
515     attachment_handler=attachment_handler_two,
516 )
517 alto_three_musicmaker_three = MusicMaker(
518     rmaker=rmaker_three,
519     pitches=alto_3_note,
520     continuous=True,
521     attachment_handler=attachment_handler_three,
522 )
523 #####alto_four#####
524 alto_four_musicmaker_one = MusicMaker(
525     rmaker=rmaker_one,
526     pitches=alto_4_chord,
527     continuous=True,
528     attachment_handler=attachment_handler_one,
529 )
530 alto_four_musicmaker_two = MusicMaker(
531     rmaker=rmaker_two,
532     pitches=alto_4_random_walk_notes,
533     continuous=True,
534     attachment_handler=attachment_handler_two,
535 )
536 alto_four_musicmaker_three = MusicMaker(
537     rmaker=rmaker_three,
538     pitches=alto_4_note,
539     continuous=True,
540     attachment_handler=attachment_handler_three,
541 )
542 #####alto_five#####
543 alto_five_musicmaker_one = MusicMaker(
544     rmaker=rmaker_one,
545     pitches=alto_5_chord,
546     continuous=True,
547     attachment_handler=attachment_handler_one,
548 )
549 alto_five_musicmaker_two = MusicMaker(
550     rmaker=rmaker_two,
551     pitches=alto_5_random_walk_notes,
552     continuous=True,
553     attachment_handler=attachment_handler_two,
554 )
555 alto_five_musicmaker_three = MusicMaker(
556     rmaker=rmaker_three,
557     pitches=alto_5_note,
558     continuous=True,
559     attachment_handler=attachment_handler_three,
560 )
561 #####alto_six#####
562 alto_six_musicmaker_one = MusicMaker(
563     rmaker=rmaker_one,
564     pitches=alto_6_chord,
565     continuous=True,
566     attachment_handler=attachment_handler_one,
```

```
567 )
568 alto_six_musicmaker_two = MusicMaker(
569     rmaker=rmaker_two,
570     pitches=alto_6_random_walk_notes,
571     continuous=True,
572     attachment_handler=attachment_handler_two,
573 )
574 alto_six_musicmaker_three = MusicMaker(
575     rmaker=rmaker_three,
576     pitches=alto_6_note,
577     continuous=True,
578     attachment_handler=attachment_handler_three,
579 )
580 #####tenor_one#####
581 tenor_one_musicmaker_one = MusicMaker(
582     rmaker=rmaker_one,
583     pitches=tenor_1_chord,
584     continuous=True,
585     attachment_handler=attachment_handler_one,
586 )
587 tenor_one_musicmaker_two = MusicMaker(
588     rmaker=rmaker_two,
589     pitches=tenor_1_random_walk_notes,
590     continuous=True,
591     attachment_handler=attachment_handler_two,
592 )
593 tenor_one_musicmaker_three = MusicMaker(
594     rmaker=rmaker_three,
595     pitches=tenor_1_note,
596     continuous=True,
597     attachment_handler=attachment_handler_three,
598 )
599 #####tenor_two#####
600 tenor_two_musicmaker_one = MusicMaker(
601     rmaker=rmaker_one,
602     pitches=tenor_2_chord,
603     continuous=True,
604     attachment_handler=attachment_handler_one,
605 )
606 tenor_two_musicmaker_two = MusicMaker(
607     rmaker=rmaker_two,
608     pitches=tenor_2_random_walk_notes,
609     continuous=True,
610     attachment_handler=attachment_handler_two,
611 )
612 tenor_two_musicmaker_three = MusicMaker(
613     rmaker=rmaker_three,
614     pitches=tenor_2_note,
615     continuous=True,
616     attachment_handler=attachment_handler_three,
617 )
618 #####tenor_three#####
619 tenor_three_musicmaker_one = MusicMaker(
620     rmaker=rmaker_one,
621     pitches=tenor_3_chord,
622     continuous=True,
623     attachment_handler=attachment_handler_one,
624 )
625 tenor_three_musicmaker_two = MusicMaker(
```

```
626     rmaker=rmaker_two,
627     pitches=tenor_3_random_walk_notes,
628     continuous=True,
629     attachment_handler=attachment_handler_two,
630 )
631 tenor_three_musicmaker_three = MusicMaker(
632     rmaker=rmaker_three,
633     pitches=tenor_3_note,
634     continuous=True,
635     attachment_handler=attachment_handler_three,
636 )
637 #####tenor_four#####
638 tenor_four_musicmaker_one = MusicMaker(
639     rmaker=rmaker_one,
640     pitches=tenor_4_chord,
641     continuous=True,
642     attachment_handler=attachment_handler_one,
643 )
644 tenor_four_musicmaker_two = MusicMaker(
645     rmaker=rmaker_two,
646     pitches=tenor_4_random_walk_notes,
647     continuous=True,
648     attachment_handler=attachment_handler_two,
649 )
650 tenor_four_musicmaker_three = MusicMaker(
651     rmaker=rmaker_three,
652     pitches=tenor_4_note,
653     continuous=True,
654     attachment_handler=attachment_handler_three,
655 )
656 #####tenor_five#####
657 tenor_five_musicmaker_one = MusicMaker(
658     rmaker=rmaker_one,
659     pitches=tenor_5_chord,
660     continuous=True,
661     attachment_handler=attachment_handler_one,
662 )
663 tenor_five_musicmaker_two = MusicMaker(
664     rmaker=rmaker_two,
665     pitches=tenor_5_random_walk_notes,
666     continuous=True,
667     attachment_handler=attachment_handler_two,
668 )
669 tenor_five_musicmaker_three = MusicMaker(
670     rmaker=rmaker_three,
671     pitches=tenor_5_note,
672     continuous=True,
673     attachment_handler=attachment_handler_three,
674 )
675 #####baritone_one#####
676 baritone_one_musicmaker_one = MusicMaker(
677     rmaker=rmaker_one,
678     pitches=baritone_1_chord,
679     continuous=True,
680     attachment_handler=attachment_handler_one,
681 )
682 baritone_one_musicmaker_two = MusicMaker(
683     rmaker=rmaker_two,
684     pitches=baritone_1_random_walk_notes,
```

```
685     continuous=True,
686     attachment_handler=attachment_handler_two,
687 )
688 baritone_one_musicmaker_three = MusicMaker(
689     rmaker=rmaker_three,
690     pitches=baritone_1_note,
691     continuous=True,
692     attachment_handler=attachment_handler_three,
693 )
694 #####baritone_two#####
695 baritone_two_musicmaker_one = MusicMaker(
696     rmaker=rmaker_one,
697     pitches=baritone_2_chord,
698     continuous=True,
699     attachment_handler=attachment_handler_one,
700 )
701 baritone_two_musicmaker_two = MusicMaker(
702     rmaker=rmaker_two,
703     pitches=baritone_2_random_walk_notes,
704     continuous=True,
705     attachment_handler=attachment_handler_two,
706 )
707 baritone_two_musicmaker_three = MusicMaker(
708     rmaker=rmaker_three,
709     pitches=baritone_2_note,
710     continuous=True,
711     attachment_handler=attachment_handler_three,
712 )
713 #####baritone_three#####
714 baritone_three_musicmaker_one = MusicMaker(
715     rmaker=rmaker_one,
716     pitches=baritone_3_chord,
717     continuous=True,
718     attachment_handler=attachment_handler_one,
719 )
720 baritone_three_musicmaker_two = MusicMaker(
721     rmaker=rmaker_two,
722     pitches=baritone_3_random_walk_notes,
723     continuous=True,
724     attachment_handler=attachment_handler_two,
725 )
726 baritone_three_musicmaker_three = MusicMaker(
727     rmaker=rmaker_three,
728     pitches=baritone_3_note,
729     continuous=True,
730     attachment_handler=attachment_handler_three,
731 )
732 #####bass_one#####
733 bass_one_musicmaker_one = MusicMaker(
734     rmaker=rmaker_one,
735     pitches=bass_1_chord,
736     continuous=True,
737     attachment_handler=attachment_handler_one,
738 )
739 bass_one_musicmaker_two = MusicMaker(
740     rmaker=rmaker_two,
741     pitches=bass_1_random_walk_notes,
742     continuous=True,
743     attachment_handler=attachment_handler_two,
```

```

744 )
745 bass_one_musicmaker_three = MusicMaker(
746     rmaker=rmaker_three,
747     pitches=bass_1_note,
748     continuous=True,
749     attachment_handler=attachment_handler_three,
750 )
751 #####bass_two#####
752 bass_two_musicmaker_one = MusicMaker(
753     rmaker=rmaker_one,
754     pitches=bass_2_chord,
755     continuous=True,
756     attachment_handler=attachment_handler_one,
757 )
758 bass_two_musicmaker_two = MusicMaker(
759     rmaker=rmaker_two,
760     pitches=bass_2_random_walk_notes,
761     continuous=True,
762     attachment_handler=attachment_handler_two,
763 )
764 bass_two_musicmaker_three = MusicMaker(
765     rmaker=rmaker_three,
766     pitches=bass_2_note,
767     continuous=True,
768     attachment_handler=attachment_handler_three,
769 )
770 #####contrabass#####
771 contrabass_musicmaker_one = MusicMaker(
772     rmaker=rmaker_one,
773     pitches=contrabass_chord,
774     continuous=True,
775     attachment_handler=attachment_handler_one,
776 )
777 contrabass_musicmaker_two = MusicMaker(
778     rmaker=rmaker_two,
779     pitches=contrabass_random_walk_notes,
780     continuous=True,
781     attachment_handler=attachment_handler_two,
782 )
783 contrabass_musicmaker_three = MusicMaker(
784     rmaker=rmaker_three,
785     pitches=contrabass_note,
786     continuous=True,
787     attachment_handler=attachment_handler_three,
788 )
789
790 silence_maker = abjadext.rmakers.NoteRhythmMaker(
791     division_masks=[
792         abjadext.rmakers.SilenceMask(
793             pattern=abjad.index([0], 1),
794             ),
795         ],
796     )
797
798 class MusicSpecifier:
799
800     def __init__(self, music_maker, voice_name):
801         self.music_maker = music_maker
802         self.voice_name = voice_name

```

```

803
804 print('Collecting timespans and rmakers ...')
805
806 voice_1_timespan_list = abjad.TimespanList([
807     abjad.AnnotatedTimespan(
808         start_offset=start_offset,
809         stop_offset=stop_offset,
810         annotation=MusicSpecifier(
811             music_maker=music_maker,
812             voice_name='Voice 1',
813         ),
814     )
815     for start_offset, stop_offset, music_maker in [
816         [(0, 8), (2, 8), soprano_musicmaker_two],
817         [(2, 8), (4, 8), soprano_musicmaker_one],
818         [(4, 8), (6, 8), soprano_musicmaker_two],
819         [(6, 8), (8, 8), soprano_musicmaker_two],
820
821         [(10, 8), (12, 8), soprano_musicmaker_two],
822         [(12, 8), (14, 8), soprano_musicmaker_two],
823         [(14, 8), (16, 8), soprano_musicmaker_two],
824         [(16, 8), (18, 8), soprano_musicmaker_one],
825         [(18, 8), (20, 8), soprano_musicmaker_two],
826         [(20, 8), (22, 8), soprano_musicmaker_two],
827         [(22, 8), (24, 8), soprano_musicmaker_two],
828
829         [(26, 8), (28, 8), soprano_musicmaker_one],
830         [(28, 8), (30, 8), soprano_musicmaker_two],
831         [(30, 8), (32, 8), soprano_musicmaker_two],
832     ]
833 ])
834
835 voice_2_timespan_list = abjad.TimespanList([
836     abjad.AnnotatedTimespan(
837         start_offset=start_offset,
838         stop_offset=stop_offset,
839         annotation=MusicSpecifier(
840             music_maker=music_maker,
841             voice_name='Voice 2',
842         ),
843     )
844     for start_offset, stop_offset, music_maker in [
845         [(0, 8), (2, 8), soprano_one_musicmaker_two],
846         [(2, 8), (4, 8), soprano_one_musicmaker_three],
847         [(4, 8), (6, 8), soprano_one_musicmaker_two],
848
849         [(8, 8), (10, 8), soprano_one_musicmaker_three],
850         [(10, 8), (12, 8), soprano_one_musicmaker_two],
851         [(12, 8), (14, 8), soprano_one_musicmaker_two],
852         [(14, 8), (16, 8), soprano_one_musicmaker_one],
853         [(16, 8), (18, 8), soprano_one_musicmaker_three],
854         [(18, 8), (20, 8), soprano_one_musicmaker_one],
855
856         [(22, 8), (24, 8), soprano_one_musicmaker_one],
857         [(24, 8), (26, 8), soprano_one_musicmaker_one],
858         [(26, 8), (28, 8), soprano_one_musicmaker_three],
859         [(28, 8), (30, 8), soprano_one_musicmaker_two],
860         [(30, 8), (32, 8), soprano_one_musicmaker_two],
861     ]

```

```

862 ])
863
864 voice_3_timespan_list = abjad.TimespanList([
865     abjad.AnnotatedTimespan(
866         start_offset=start_offset,
867         stop_offset=stop_offset,
868         annotation=MusicSpecifier(
869             music_maker=music_maker,
870             voice_name='Voice 3',
871         ),
872     )
873     for start_offset, stop_offset, music_maker in [
874         [(0, 8), (2, 8), soprano_two_musicmaker_one],
875         [(2, 8), (4, 8), soprano_two_musicmaker_one],
876         [(4, 8), (6, 8), soprano_two_musicmaker_one],
877         [(6, 8), (8, 8), soprano_two_musicmaker_one],
878         [(8, 8), (10, 8), soprano_two_musicmaker_one],
879         [(10, 8), (12, 8), soprano_two_musicmaker_three],
880         [(12, 8), (14, 8), soprano_two_musicmaker_one],
881         [(14, 8), (16, 8), soprano_two_musicmaker_one],
882         [(16, 8), (18, 8), soprano_two_musicmaker_two],
883
884         [(20, 8), (22, 8), soprano_two_musicmaker_two],
885         [(22, 8), (24, 8), soprano_two_musicmaker_three],
886         [(24, 8), (26, 8), soprano_two_musicmaker_two],
887         [(26, 8), (28, 8), soprano_two_musicmaker_two],
888         [(28, 8), (30, 8), soprano_two_musicmaker_two],
889         [(30, 8), (32, 8), soprano_two_musicmaker_one],
890     ]
891 ])
892
893 voice_4_timespan_list = abjad.TimespanList([
894     abjad.AnnotatedTimespan(
895         start_offset=start_offset,
896         stop_offset=stop_offset,
897         annotation=MusicSpecifier(
898             music_maker=music_maker,
899             voice_name='Voice 4',
900         ),
901     )
902     for start_offset, stop_offset, music_maker in [
903         [(0, 8), (2, 8), soprano_three_musicmaker_two],
904         [(2, 8), (4, 8), soprano_three_musicmaker_two],
905         [(4, 8), (6, 8), soprano_three_musicmaker_two],
906         [(6, 8), (8, 8), soprano_three_musicmaker_three],
907         [(8, 8), (10, 8), soprano_three_musicmaker_one],
908         [(10, 8), (12, 8), soprano_three_musicmaker_one],
909         [(12, 8), (14, 8), soprano_three_musicmaker_one],
910         [(14, 8), (16, 8), soprano_three_musicmaker_three],
911         [(16, 8), (18, 8), soprano_three_musicmaker_two],
912
913         [(20, 8), (22, 8), soprano_three_musicmaker_two],
914         [(22, 8), (24, 8), soprano_three_musicmaker_three],
915         [(24, 8), (26, 8), soprano_three_musicmaker_two],
916         [(26, 8), (28, 8), soprano_three_musicmaker_two],
917         [(28, 8), (30, 8), soprano_three_musicmaker_two],
918         [(30, 8), (32, 8), soprano_three_musicmaker_three],
919     ]
920 ])

```

```

921
922 voice_5_timespan_list = abjad.TimespanList([
923     abjad.AnnotatedTimespan(
924         start_offset=start_offset,
925         stop_offset=stop_offset,
926         annotation=MusicSpecifier(
927             music_maker=music_maker,
928             voice_name='Voice 5',
929         ),
930     )
931     for start_offset, stop_offset, music_maker in [
932         [(0, 8), (2, 8), alto_one_musicmaker_one],
933         [(2, 8), (4, 8), alto_one_musicmaker_one],
934         [(4, 8), (6, 8), alto_one_musicmaker_two],
935         [(6, 8), (8, 8), alto_one_musicmaker_two],
936         [(8, 8), (10, 8), alto_one_musicmaker_three],
937
938         [(12, 8), (14, 8), alto_one_musicmaker_two],
939         [(14, 8), (16, 8), alto_one_musicmaker_one],
940         [(16, 8), (18, 8), alto_one_musicmaker_one],
941         [(18, 8), (20, 8), alto_one_musicmaker_three],
942
943         [(22, 8), (24, 8), alto_one_musicmaker_one],
944         [(24, 8), (26, 8), alto_one_musicmaker_two],
945         [(26, 8), (28, 8), alto_one_musicmaker_two],
946         [(28, 8), (30, 8), alto_one_musicmaker_three],
947         [(30, 8), (32, 8), alto_one_musicmaker_two],
948     ]
949 ])
950
951 voice_6_timespan_list = abjad.TimespanList([
952     abjad.AnnotatedTimespan(
953         start_offset=start_offset,
954         stop_offset=stop_offset,
955         annotation=MusicSpecifier(
956             music_maker=music_maker,
957             voice_name='Voice 6',
958         ),
959     )
960     for start_offset, stop_offset, music_maker in [
961         [(0, 8), (2, 8), alto_two_musicmaker_one],
962         [(2, 8), (4, 8), alto_two_musicmaker_one],
963         [(4, 8), (6, 8), alto_two_musicmaker_one],
964         [(6, 8), (8, 8), alto_two_musicmaker_one],
965         [(8, 8), (10, 8), alto_two_musicmaker_two],
966         [(10, 8), (12, 8), alto_two_musicmaker_three],
967
968         [(14, 8), (16, 8), alto_two_musicmaker_two],
969         [(16, 8), (18, 8), alto_two_musicmaker_two],
970         [(18, 8), (20, 8), alto_two_musicmaker_two],
971         [(20, 8), (22, 8), alto_two_musicmaker_one],
972         [(22, 8), (24, 8), alto_two_musicmaker_three],
973         [(24, 8), (26, 8), alto_two_musicmaker_one],
974
975         [(28, 8), (30, 8), alto_two_musicmaker_one],
976         [(30, 8), (32, 8), alto_two_musicmaker_two],
977     ]
978 ])
979

```

```

980 voice_7_timespan_list = abjad.TimespanList([
981     abjad.AnnotatedTimespan(
982         start_offset=start_offset,
983         stop_offset=stop_offset,
984         annotation=MusicSpecifier(
985             music_maker=music_maker,
986             voice_name='Voice 7',
987         ),
988     ),
989     for start_offset, stop_offset, music_maker in [
990         [(0, 8), (2, 8), alto_three_musicmaker_two],
991
992         [(4, 8), (6, 8), alto_three_musicmaker_two],
993         [(6, 8), (8, 8), alto_three_musicmaker_two],
994         [(8, 8), (10, 8), alto_three_musicmaker_two],
995
996         [(12, 8), (14, 8), alto_three_musicmaker_two],
997         [(14, 8), (16, 8), alto_three_musicmaker_one],
998         [(16, 8), (18, 8), alto_three_musicmaker_three],
999         [(18, 8), (20, 8), alto_three_musicmaker_one],
1000
1001         [(22, 8), (24, 8), alto_three_musicmaker_two],
1002         [(24, 8), (26, 8), alto_three_musicmaker_two],
1003         [(26, 8), (28, 8), alto_three_musicmaker_two],
1004     ]
1005 ])
1006
1007 voice_8_timespan_list = abjad.TimespanList([
1008     abjad.AnnotatedTimespan(
1009         start_offset=start_offset,
1010         stop_offset=stop_offset,
1011         annotation=MusicSpecifier(
1012             music_maker=music_maker,
1013             voice_name='Voice 8',
1014         ),
1015     )
1016     for start_offset, stop_offset, music_maker in [
1017         [(0, 8), (2, 8), alto_four_musicmaker_two],
1018         [(2, 8), (4, 8), alto_four_musicmaker_two],
1019         [(4, 8), (6, 8), alto_four_musicmaker_two],
1020         [(6, 8), (8, 8), alto_four_musicmaker_one],
1021         [(8, 8), (10, 8), alto_four_musicmaker_one],
1022         [(10, 8), (12, 8), alto_four_musicmaker_one],
1023         [(12, 8), (14, 8), alto_four_musicmaker_two],
1024
1025         [(16, 8), (18, 8), alto_four_musicmaker_two],
1026         [(18, 8), (20, 8), alto_four_musicmaker_one],
1027         [(20, 8), (22, 8), alto_four_musicmaker_three],
1028         [(22, 8), (24, 8), alto_four_musicmaker_one],
1029         [(24, 8), (26, 8), alto_four_musicmaker_one],
1030         [(26, 8), (28, 8), alto_four_musicmaker_two],
1031         [(28, 8), (30, 8), alto_four_musicmaker_two],
1032     ]
1033 ])
1034
1035 voice_9_timespan_list = abjad.TimespanList([
1036     abjad.AnnotatedTimespan(
1037         start_offset=start_offset,
1038         stop_offset=stop_offset,

```

```

1039     annotation=MusicSpecifier(
1040         music_maker=music_maker,
1041         voice_name='Voice 9',
1042     ),
1043 )
1044 for start_offset, stop_offset, music_maker in [
1045     [(0, 8), (2, 8), alto_five_musicmaker_two],
1046     [(2, 8), (4, 8), alto_five_musicmaker_one],
1047     [(4, 8), (6, 8), alto_five_musicmaker_one],
1048     [(6, 8), (8, 8), alto_five_musicmaker_two],
1049     [(8, 8), (10, 8), alto_five_musicmaker_three],
1050
1051     [(12, 8), (14, 8), alto_five_musicmaker_one],
1052     [(14, 8), (16, 8), alto_five_musicmaker_one],
1053     [(16, 8), (18, 8), alto_five_musicmaker_one],
1054     [(18, 8), (20, 8), alto_five_musicmaker_one],
1055     [(20, 8), (22, 8), alto_five_musicmaker_three],
1056     [(22, 8), (24, 8), alto_five_musicmaker_two],
1057     [(24, 8), (26, 8), alto_five_musicmaker_two],
1058
1059     [(28, 8), (30, 8), alto_five_musicmaker_two],
1060     [(30, 8), (32, 8), alto_five_musicmaker_three],
1061 ]
1062 ])
1063
1064 voice_10_timespan_list = abjad.TimespanList([
1065     abjad.AnnotatedTimespan(
1066         start_offset=start_offset,
1067         stop_offset=stop_offset,
1068         annotation=MusicSpecifier(
1069             music_maker=music_maker,
1070             voice_name='Voice 10',
1071         ),
1072     ),
1073     for start_offset, stop_offset, music_maker in [
1074         [(0, 8), (2, 8), alto_six_musicmaker_one],
1075
1076         [(4, 8), (6, 8), alto_six_musicmaker_three],
1077         [(6, 8), (8, 8), alto_six_musicmaker_one],
1078         [(8, 8), (10, 8), alto_six_musicmaker_two],
1079         [(10, 8), (12, 8), alto_six_musicmaker_three],
1080         [(12, 8), (14, 8), alto_six_musicmaker_two],
1081         [(14, 8), (16, 8), alto_six_musicmaker_two],
1082         [(16, 8), (18, 8), alto_six_musicmaker_one],
1083
1084         [(20, 8), (22, 8), alto_six_musicmaker_one],
1085         [(22, 8), (24, 8), alto_six_musicmaker_one],
1086         [(24, 8), (26, 8), alto_six_musicmaker_one],
1087         [(26, 8), (28, 8), alto_six_musicmaker_three],
1088         [(28, 8), (30, 8), alto_six_musicmaker_two],
1089         [(30, 8), (32, 8), alto_six_musicmaker_two],
1090     ]
1091 ])
1092
1093 voice_11_timespan_list = abjad.TimespanList([
1094     abjad.AnnotatedTimespan(
1095         start_offset=start_offset,
1096         stop_offset=stop_offset,
1097         annotation=MusicSpecifier(

```

```

1098     music_maker=music_maker,
1099     voice_name='Voice 11',
1100   ),
1101 )
1102 for start_offset, stop_offset, music_maker in [
1103   [(0, 8), (2, 8), tenor_one_musicmaker_one],
1104   [(2, 8), (4, 8), tenor_one_musicmaker_one],
1105   [(4, 8), (6, 8), tenor_one_musicmaker_two],
1106   [(6, 8), (8, 8), tenor_one_musicmaker_two],
1107   [(8, 8), (10, 8), tenor_one_musicmaker_three],
1108   [(10, 8), (12, 8), tenor_one_musicmaker_one],
1109   [(12, 8), (14, 8), tenor_one_musicmaker_one],
1110
1111   [(16, 8), (18, 8), tenor_one_musicmaker_one],
1112   [(18, 8), (20, 8), tenor_one_musicmaker_two],
1113   [(20, 8), (22, 8), tenor_one_musicmaker_two],
1114   [(22, 8), (24, 8), tenor_one_musicmaker_three],
1115   [(24, 8), (26, 8), tenor_one_musicmaker_two],
1116   [(26, 8), (28, 8), tenor_one_musicmaker_one],
1117   [(28, 8), (30, 8), tenor_one_musicmaker_one],
1118   [(30, 8), (32, 8), tenor_one_musicmaker_one],
1119 ]
1120 ])
1121
1122 voice_12_timestrap_list = abjad.TimespanList([
1123   abjad.AnnotatedTimespan(
1124     start_offset=start_offset,
1125     stop_offset=stop_offset,
1126     annotation=MusicSpecifier(
1127       music_maker=music_maker,
1128       voice_name='Voice 12',
1129     ),
1130   ),
1131   for start_offset, stop_offset, music_maker in [
1132     [(0, 8), (2, 8), tenor_two_musicmaker_two],
1133     [(2, 8), (4, 8), tenor_two_musicmaker_one],
1134     [(4, 8), (6, 8), tenor_two_musicmaker_one],
1135     [(6, 8), (8, 8), tenor_two_musicmaker_one],
1136     [(8, 8), (10, 8), tenor_two_musicmaker_two],
1137     [(10, 8), (12, 8), tenor_two_musicmaker_one],
1138     [(12, 8), (14, 8), tenor_two_musicmaker_one],
1139     [(14, 8), (16, 8), tenor_two_musicmaker_two],
1140     [(16, 8), (18, 8), tenor_two_musicmaker_three],
1141     [(18, 8), (20, 8), tenor_two_musicmaker_two],
1142
1143     [(22, 8), (24, 8), tenor_two_musicmaker_two],
1144     [(24, 8), (26, 8), tenor_two_musicmaker_two],
1145     [(26, 8), (28, 8), tenor_two_musicmaker_two],
1146     [(28, 8), (30, 8), tenor_two_musicmaker_one],
1147     [(30, 8), (32, 8), tenor_two_musicmaker_one],
1148   ]
1149 ])
1150
1151 voice_13_timestrap_list = abjad.TimespanList([
1152   abjad.AnnotatedTimespan(
1153     start_offset=start_offset,
1154     stop_offset=stop_offset,
1155     annotation=MusicSpecifier(
1156       music_maker=music_maker,

```

```

1157     voice_name='Voice 13',
1158     ),
1159   )
1160   for start_offset, stop_offset, music_maker in [
1161     [(0, 8), (2, 8), tenor_three_musicmaker_three],
1162     [(2, 8), (4, 8), tenor_three_musicmaker_one],
1163     [(4, 8), (6, 8), tenor_three_musicmaker_three],
1164     [(6, 8), (8, 8), tenor_three_musicmaker_one],
1165     [(8, 8), (10, 8), tenor_three_musicmaker_one],
1166     [(10, 8), (12, 8), tenor_three_musicmaker_three],
1167     [(12, 8), (14, 8), tenor_three_musicmaker_two],
1168     [(14, 8), (16, 8), tenor_three_musicmaker_two],
1169
1170     [(18, 8), (20, 8), tenor_three_musicmaker_three],
1171     [(20, 8), (22, 8), tenor_three_musicmaker_two],
1172     [(22, 8), (24, 8), tenor_three_musicmaker_two],
1173     [(24, 8), (26, 8), tenor_three_musicmaker_one],
1174     [(26, 8), (28, 8), tenor_three_musicmaker_one],
1175     [(28, 8), (30, 8), tenor_three_musicmaker_three],
1176     [(30, 8), (32, 8), tenor_three_musicmaker_two],
1177   ]
1178 ])
1179
1180 voice_14_timespan_list = abjad.TimespanList([
1181   abjad.AnnotatedTimespan(
1182     start_offset=start_offset,
1183     stop_offset=stop_offset,
1184     annotation=MusicSpecifier(
1185       music_maker=music_maker,
1186       voice_name='Voice 14',
1187     ),
1188   ),
1189   for start_offset, stop_offset, music_maker in [
1190     [(0, 8), (2, 8), tenor_four_musicmaker_two],
1191     [(2, 8), (4, 8), tenor_four_musicmaker_two],
1192     [(4, 8), (6, 8), tenor_four_musicmaker_two],
1193     [(6, 8), (8, 8), tenor_four_musicmaker_two],
1194     [(8, 8), (10, 8), tenor_four_musicmaker_two],
1195     [(10, 8), (12, 8), tenor_four_musicmaker_one],
1196     [(12, 8), (14, 8), tenor_four_musicmaker_two],
1197     [(14, 8), (16, 8), tenor_four_musicmaker_two],
1198     [(16, 8), (18, 8), tenor_four_musicmaker_one],
1199
1200     [(22, 8), (24, 8), tenor_four_musicmaker_one],
1201     [(24, 8), (26, 8), tenor_four_musicmaker_three],
1202     [(26, 8), (28, 8), tenor_four_musicmaker_one],
1203     [(28, 8), (30, 8), tenor_four_musicmaker_one],
1204     [(30, 8), (32, 8), tenor_four_musicmaker_one],
1205   ]
1206 ])
1207
1208 voice_15_timespan_list = abjad.TimespanList([
1209   abjad.AnnotatedTimespan(
1210     start_offset=start_offset,
1211     stop_offset=stop_offset,
1212     annotation=MusicSpecifier(
1213       music_maker=music_maker,
1214       voice_name='Voice 15',
1215     ),

```

```

1216     )
1217     for start_offset, stop_offset, music_maker in [
1218         [(0, 8), (2, 8), tenor_five_musicmaker_two],
1219         [(2, 8), (4, 8), tenor_five_musicmaker_two],
1220         [(4, 8), (6, 8), tenor_five_musicmaker_two],
1221         [(6, 8), (8, 8), tenor_five_musicmaker_one],
1222         [(8, 8), (10, 8), tenor_five_musicmaker_three],
1223         [(10, 8), (12, 8), tenor_five_musicmaker_one],
1224
1225         [(14, 8), (16, 8), tenor_five_musicmaker_two],
1226         [(16, 8), (18, 8), tenor_five_musicmaker_two],
1227         [(18, 8), (20, 8), tenor_five_musicmaker_three],
1228         [(20, 8), (22, 8), tenor_five_musicmaker_one],
1229         [(22, 8), (24, 8), tenor_five_musicmaker_one],
1230         [(24, 8), (26, 8), tenor_five_musicmaker_one],
1231         [(26, 8), (28, 8), tenor_five_musicmaker_one],
1232         [(28, 8), (30, 8), tenor_five_musicmaker_three],
1233     ]
1234 ])
1235
1236 voice_16_timespan_list = abjad.TimespanList([
1237     abjad.AnnotatedTimespan(
1238         start_offset=start_offset,
1239         stop_offset=stop_offset,
1240         annotation=MusicSpecifier(
1241             music_maker=music_maker,
1242             voice_name='Voice 16',
1243         ),
1244     )
1245     for start_offset, stop_offset, music_maker in [
1246         [(0, 8), (2, 8), baritone_one_musicmaker_two],
1247         [(2, 8), (4, 8), baritone_one_musicmaker_two],
1248         [(4, 8), (6, 8), baritone_one_musicmaker_two],
1249
1250         [(8, 8), (10, 8), baritone_one_musicmaker_one],
1251         [(10, 8), (12, 8), baritone_one_musicmaker_three],
1252         [(12, 8), (14, 8), baritone_one_musicmaker_one],
1253         [(14, 8), (16, 8), baritone_one_musicmaker_two],
1254
1255         [(18, 8), (20, 8), baritone_one_musicmaker_two],
1256         [(20, 8), (22, 8), baritone_one_musicmaker_two],
1257         [(22, 8), (24, 8), baritone_one_musicmaker_one],
1258         [(24, 8), (26, 8), baritone_one_musicmaker_one],
1259
1260         [(28, 8), (30, 8), baritone_one_musicmaker_two],
1261         [(30, 8), (32, 8), baritone_one_musicmaker_two],
1262     ]
1263 ])
1264
1265 voice_17_timespan_list = abjad.TimespanList([
1266     abjad.AnnotatedTimespan(
1267         start_offset=start_offset,
1268         stop_offset=stop_offset,
1269         annotation=MusicSpecifier(
1270             music_maker=music_maker,
1271             voice_name='Voice 17',
1272         ),
1273     )
1274     for start_offset, stop_offset, music_maker in [

```

```

1275     [(0, 8), (2, 8), baritone_two_musicmaker_one],
1276     [(2, 8), (4, 8), baritone_two_musicmaker_one],
1277     [(4, 8), (6, 8), baritone_two_musicmaker_two],
1278     [(6, 8), (8, 8), baritone_two_musicmaker_two],
1279     [(8, 8), (10, 8), baritone_two_musicmaker_two],
1280     [(10, 8), (12, 8), baritone_two_musicmaker_three],
1281     [(12, 8), (14, 8), baritone_two_musicmaker_one],
1282     [(14, 8), (16, 8), baritone_two_musicmaker_one],
1283     [(16, 8), (18, 8), baritone_two_musicmaker_one],
1284
1285     [(22, 8), (24, 8), baritone_two_musicmaker_two],
1286     [(24, 8), (26, 8), baritone_two_musicmaker_two],
1287     [(26, 8), (28, 8), baritone_two_musicmaker_two],
1288     [(28, 8), (30, 8), baritone_two_musicmaker_three],
1289     [(30, 8), (32, 8), baritone_two_musicmaker_one],
1290 ]
1291 ])
1292
1293 voice_18_timespan_list = abjad.TimespanList([
1294     abjad.AnnotatedTimespan(
1295         start_offset=start_offset,
1296         stop_offset=stop_offset,
1297         annotation=MusicSpecifier(
1298             music_maker=music_maker,
1299             voice_name='Voice 18',
1300         ),
1301     )
1302     for start_offset, stop_offset, music_maker in [
1303         [(0, 8), (2, 8), baritone_three_musicmaker_one],
1304         [(2, 8), (4, 8), baritone_three_musicmaker_one],
1305         [(4, 8), (6, 8), baritone_three_musicmaker_one],
1306         [(6, 8), (8, 8), baritone_three_musicmaker_one],
1307         [(8, 8), (10, 8), baritone_three_musicmaker_three],
1308         [(10, 8), (12, 8), baritone_three_musicmaker_two],
1309
1310         [(16, 8), (18, 8), baritone_three_musicmaker_one],
1311
1312         [(20, 8), (22, 8), baritone_three_musicmaker_one],
1313         [(22, 8), (24, 8), baritone_three_musicmaker_one],
1314         [(24, 8), (26, 8), baritone_three_musicmaker_three],
1315         [(26, 8), (28, 8), baritone_three_musicmaker_two],
1316         [(28, 8), (30, 8), baritone_three_musicmaker_two],
1317         [(30, 8), (32, 8), baritone_three_musicmaker_two],
1318     ]
1319 ])
1320
1321 voice_19_timespan_list = abjad.TimespanList([
1322     abjad.AnnotatedTimespan(
1323         start_offset=start_offset,
1324         stop_offset=stop_offset,
1325         annotation=MusicSpecifier(
1326             music_maker=music_maker,
1327             voice_name='Voice 19',
1328         ),
1329     )
1330     for start_offset, stop_offset, music_maker in [
1331         [(0, 8), (2, 8), bass_one_musicmaker_two],
1332         [(2, 8), (4, 8), bass_one_musicmaker_two],
1333         [(4, 8), (6, 8), bass_one_musicmaker_two],

```

```

1334     [(6, 8), (8, 8), bass_one_musicmaker_one],
1335     [(8, 8), (10, 8), bass_one_musicmaker_one],
1336     [(10, 8), (12, 8), bass_one_musicmaker_two],
1337     [(12, 8), (14, 8), bass_one_musicmaker_two],
1338     [(14, 8), (16, 8), bass_one_musicmaker_two],
1339     [(16, 8), (18, 8), bass_one_musicmaker_three],
1340     [(18, 8), (20, 8), bass_one_musicmaker_one],
1341
1342     [(22, 8), (24, 8), bass_one_musicmaker_two],
1343     [(24, 8), (26, 8), bass_one_musicmaker_two],
1344     [(26, 8), (28, 8), bass_one_musicmaker_two],
1345     [(28, 8), (30, 8), bass_one_musicmaker_three],
1346     [(30, 8), (32, 8), bass_one_musicmaker_one],
1347 ]
1348 ])
1349
1350 voice_20_timespan_list = abjad.TimespanList([
1351     abjad.AnnotatedTimespan(
1352         start_offset=start_offset,
1353         stop_offset=stop_offset,
1354         annotation=MusicSpecifier(
1355             music_maker=music_maker,
1356             voice_name='Voice 20',
1357         ),
1358     )
1359     for start_offset, stop_offset, music_maker in [
1360         [(0, 8), (2, 8), bass_two_musicmaker_two],
1361         [(2, 8), (4, 8), bass_two_musicmaker_two],
1362         [(4, 8), (6, 8), bass_two_musicmaker_two],
1363         [(6, 8), (8, 8), bass_two_musicmaker_one],
1364         [(8, 8), (10, 8), bass_two_musicmaker_one],
1365         [(10, 8), (12, 8), bass_two_musicmaker_two],
1366         [(12, 8), (14, 8), bass_two_musicmaker_two],
1367
1368         [(16, 8), (18, 8), bass_two_musicmaker_two],
1369         [(18, 8), (20, 8), bass_two_musicmaker_one],
1370         [(20, 8), (22, 8), bass_two_musicmaker_three],
1371         [(22, 8), (24, 8), bass_two_musicmaker_two],
1372         [(24, 8), (26, 8), bass_two_musicmaker_two],
1373         [(26, 8), (28, 8), bass_two_musicmaker_two],
1374         [(28, 8), (30, 8), bass_two_musicmaker_two],
1375         [(30, 8), (32, 8), bass_two_musicmaker_three],
1376     ]
1377 ])
1378
1379 voice_21_timespan_list = abjad.TimespanList([
1380     abjad.AnnotatedTimespan(
1381         start_offset=start_offset,
1382         stop_offset=stop_offset,
1383         annotation=MusicSpecifier(
1384             music_maker=music_maker,
1385             voice_name='Voice 21',
1386         ),
1387     )
1388     for start_offset, stop_offset, music_maker in [
1389         [(0, 8), (2, 8), contrabass_musicmaker_two],
1390         [(2, 8), (4, 8), contrabass_musicmaker_two],
1391         [(4, 8), (6, 8), contrabass_musicmaker_two],
1392         [(6, 8), (8, 8), contrabass_musicmaker_three],

```



```

1452 timespan_list.sort()
1453
1454 for voice_name, timespan_list in all_timespan_lists.items():
1455     shards = timespan_list.split_at_offsets(bounds)
1456     split_timespan_list = abjad.TimespanList()
1457     for shard in shards:
1458         split_timespan_list.extend(shard)
1459     split_timespan_list.sort()
1460     all_timespan_lists[voice_name] = timespan_list
1461
1462 score = abjad.Score([
1463     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
1464     abjad.StaffGroup(
1465         [
1466             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
1467                         lilypond_type='Staff',),
1468             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
1469                         lilypond_type='Staff',),
1470             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
1471                         lilypond_type='Staff',),
1472             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
1473                         lilypond_type='Staff',),
1474             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
1475                         lilypond_type='Staff',),
1476             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
1477                         lilypond_type='Staff',),
1478             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
1479                         lilypond_type='Staff',),
1480             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
1481                         lilypond_type='Staff',),
1482             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
1483                         lilypond_type='Staff',),
1484             abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
1485                         lilypond_type='Staff',),
1486             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
1487                         lilypond_type='Staff',),
1488             abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
1489                         lilypond_type='Staff',),
1490             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
1491                         lilypond_type='Staff',),
1492             abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
1493                         lilypond_type='Staff',),
1494             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
1495                         lilypond_type='Staff',),
1496             abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
1497                         lilypond_type='Staff',),
1498             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
1499                         lilypond_type='Staff',),
1500             abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
1501                         lilypond_type='Staff',),
1502             abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
1503                         lilypond_type='Staff',),
1504             abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
1505                         lilypond_type='Staff',),
1506             abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
1507                         lilypond_type='Staff',),
1508         ],
1509         name='Staff Group',
1510     )

```

```

1490 ])
1491
1492 for time_signature in time_signatures:
1493     skip = abjad.Skip(1, multiplier=(time_signature))
1494     abjad.attach(time_signature, skip)
1495     score['Global Context'].append(skip)
1496
1497 print('Making containers ...')
1498
1499 def make_container(music_maker, durations):
1500     selections = music_maker(durations)
1501     container = abjad.Container([])
1502     container.extend(selections)
1503     return container
1504
1505 def key_function(timespan):
1506     return timespan.annotation.music_maker or silence_maker
1507
1508 for voice_name, timespan_list in all_timespan_lists.items():
1509     for music_maker, grouper in itertools.groupby(
1510         timespan_list,
1511         key=key_function,
1512     ):
1513         durations = [timespan.duration for timespan in grouper]
1514         container = make_container(music_maker, durations)
1515         voice = score[voice_name]
1516         voice.append(container)
1517
1518 print('Splitting and rewriting ...')
1519 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
1520     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
1521         time_signature = time_signatures[i]
1522         abjad.mutate(shard).rewrite_meter(time_signature)
1523
1524 print('Beaming runs ...')
1525 for voice in abjad.select(score).components(abjad.Voice):
1526     for run in abjad.select(voice).runs():
1527         specifier = abjadext.rmakers.BeamSpecifier(
1528             beam_each_division=False,
1529         )
1530         specifier(run)
1531     abjad.beam(voice[:], beam_lone_notes=False, beam_rests=False)
1532
1533 print('Stopping Hairpins ...')
1534 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1535     for rest in abjad.iterate(staff).components(abjad.Rest):
1536         previous_leaf = abjad.inspect(rest).leaf(-1)
1537         if isinstance(previous_leaf, abjad.Note):
1538             abjad.attach(abjad.StopHairpin(), rest)
1539         elif isinstance(previous_leaf, abjad.Chord):
1540             abjad.attach(abjad.StopHairpin(), rest)
1541         elif isinstance(previous_leaf, abjad.Rest):
1542             pass
1543
1544 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1545     first_leaf = abjad.select(staff).leaves()[0]
1546     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
1547     abjad.attach(stop, first_leaf)
1548

```

```

1549 print('Adding attachments ...')
1550 bar_line = abjad.BarLine('||')
1551 markup = abjad.Markup(r'\bold { C }')
1552 mark = abjad.RehearsalMark(markup=markup)
1553
1554 instruments = cyc([
1555     abjad.SopraninoSaxophone(),
1556     abjad.SopranoSaxophone(),
1557     abjad.SopranoSaxophone(),
1558     abjad.SopranoSaxophone(),
1559     abjad.AltoSaxophone(),
1560     abjad.AltoSaxophone(),
1561     abjad.AltoSaxophone(),
1562     abjad.AltoSaxophone(),
1563     abjad.AltoSaxophone(),
1564     abjad.AltoSaxophone(),
1565     abjad.TenorSaxophone(),
1566     abjad.TenorSaxophone(),
1567     abjad.TenorSaxophone(),
1568     abjad.TenorSaxophone(),
1569     abjad.TenorSaxophone(),
1570     abjad.BaritoneSaxophone(),
1571     abjad.BaritoneSaxophone(),
1572     abjad.BaritoneSaxophone(),
1573     abjad.BassSaxophone(),
1574     abjad.BassSaxophone(),
1575     abjad.ContrabassSaxophone(),
1576 ])
1577
1578 abbreviations = cyc([
1579     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
1580     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
1581     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
1582     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
1583     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
1584     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
1585     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
1586     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
1587     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
1588     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
1589     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
1590     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
1591     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
1592     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
1593     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
1594     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
1595     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
1596     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
1597     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
1598     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
1599     abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
1600 ])
1601
1602 names = cyc([
1603     abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
1604     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
1605     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
1606     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
1607     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),

```

```

1608 abjad.StartMarkup(markup=abjad.Markup('Alto 2')),,
1609 abjad.StartMarkup(markup=abjad.Markup('Alto 3')),,
1610 abjad.StartMarkup(markup=abjad.Markup('Alto 4')),,
1611 abjad.StartMarkup(markup=abjad.Markup('Alto 5')),,
1612 abjad.StartMarkup(markup=abjad.Markup('Alto 6')),,
1613 abjad.StartMarkup(markup=abjad.Markup('Tenor 1')),,
1614 abjad.StartMarkup(markup=abjad.Markup('Tenor 2')),,
1615 abjad.StartMarkup(markup=abjad.Markup('Tenor 3')),,
1616 abjad.StartMarkup(markup=abjad.Markup('Tenor 4')),,
1617 abjad.StartMarkup(markup=abjad.Markup('Tenor 5')),,
1618 abjad.StartMarkup(markup=abjad.Markup('Baritone 1')),,
1619 abjad.StartMarkup(markup=abjad.Markup('Baritone 2')),,
1620 abjad.StartMarkup(markup=abjad.Markup('Baritone 3')),,
1621 abjad.StartMarkup(markup=abjad.Markup('Bass 1')),,
1622 abjad.StartMarkup(markup=abjad.Markup('Bass 2')),,
1623 abjad.StartMarkup(markup=abjad.Markup('Contrabass')),,
1624 ])
1625
1626 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1627     leaf1 = abjad.select(staff).leaves()[0]
1628     abjad.attach(next(instruments), leaf1)
1629     abjad.attach(next(abbreviations), leaf1)
1630     abjad.attach(next(names), leaf1)
1631
1632 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
1633     leaf1 = abjad.select(staff).leaves()[0]
1634     last_leaf = abjad.select(staff).leaves()[-1]
1635     abjad.attach(bar_line, last_leaf)
1636
1637 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
1638     leaf1 = abjad.select(staff).leaves()[0]
1639     abjad.attach(mark, leaf1)
1640
1641 score_file = abjad.LilyPondFile.new(
1642     score,
1643     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1644     _stylesheets/abjad.ily'],
1645 )
1646
1647 abjad.SegmentMaker.comment_measure_numbers(score)
1648 ######
1649 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_C'
1650 pdf_path = f'{directory}/Section_C.pdf'
1651 path = pathlib.Path('Section_C.pdf')
1652 if path.exists():
1653     print(f'Removing {pdf_path} ...')
1654     path.unlink()
1655 time_1 = time.time()
1656 print(f'Persisting {pdf_path} ...')
1657 result = abjad.persist(score_file).as_pdf(pdf_path)
1658 print(result[0])
1659 print(result[1])
1660 print(result[2])
1661 success = result[3]
1662 if success is False:
1663     print('LilyPond failed!')
1664 time_2 = time.time()
1665 total_time = time_2 - time_1

```

```

1666 print(f'Total time: {total_time} seconds')
1667 if path.exists():
1668     print(f'Opening {pdf_path} ...')
1669     os.system(f'open {pdf_path}')
1670 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_C/Section_C.ly').readlines()
1671 open('/Users/evansdsg2/Scores/guerrero/Build/Section_C.ly', 'w').writelines(score_lines[15:-1])
1672
1673 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
1674     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1675     signature_copy = abjad.mutate(signatures).copy()
1676     staff_copy = abjad.mutate(staff).copy()
1677     part = abjad.Score()
1678     part.insert(0, staff)
1679     part.insert(0, signature_copy)
1680     part_file = abjad.LilyPondFile.new(
1681         part,
1682         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/_stylesheets/abjad.ily'],
1683         )
1684     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano'
1685     pdf_path = f'{directory}/Section_C.pdf'
1686     path = pathlib.Path('Section_C.pdf')
1687     if path.exists():
1688         print(f'Removing {pdf_path} ...')
1689         path.unlink()
1690     time_1 = time.time()
1691     print(f'Persisting {pdf_path} ...')
1692     result = abjad.persist(part_file).as_pdf(pdf_path)
1693     print(result[0])
1694     print(result[1])
1695     print(result[2])
1696     success = result[3]
1697     if success is False:
1698         print('LilyPond failed!')
1699     time_2 = time.time()
1700     total_time = time_2 - time_1
1701     print(f'Total time: {total_time} seconds')
1702     if path.exists():
1703         print(f'Opening {pdf_path} ...')
1704         os.system(f'open {pdf_path}')
1705     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano/Section_C.ly').readlines()
1706     open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano/Section_C.ly', 'w').writelines(part_lines[15:-1])
1707
1708 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
1709     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1710     signature_copy = abjad.mutate(signatures).copy()
1711     staff_copy = abjad.mutate(staff).copy()
1712     part = abjad.Score()
1713     part.insert(0, staff)
1714     part.insert(0, signature_copy)
1715     part_file = abjad.LilyPondFile.new(
1716         part,
1717         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/_stylesheets/abjad.ily'],
1718         )

```

```

1719 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1'
1720 pdf_path = f'{directory}/Section_C.pdf'
1721 path = pathlib.Path('Section_C.pdf')
1722 if path.exists():
1723     print(f'Removing {pdf_path} ...')
1724     path.unlink()
1725 time_1 = time.time()
1726 print(f'Persisting {pdf_path} ...')
1727 result = abjad.persist(part_file).as_pdf(pdf_path)
1728 print(result[0])
1729 print(result[1])
1730 print(result[2])
1731 success = result[3]
1732 if success is False:
1733     print('LilyPond failed!')
1734 time_2 = time.time()
1735 total_time = time_2 - time_1
1736 print(f'Total time: {total_time} seconds')
1737 if path.exists():
1738     print(f'Opening {pdf_path} ...')
1739     os.system(f'open {pdf_path}')
1740 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/
Section_C.ly').readlines()
1741 open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/Section_C.ly',
'w').writelines(part_lines[15:-1])
1742
1743 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
1744     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1745     signature_copy = abjad.mutate(signatures).copy()
1746     staff_copy = abjad.mutate(staff).copy()
1747     part = abjad.Score()
1748     part.insert(0, staff)
1749     part.insert(0, signature_copy)
1750     part_file = abjad.LilyPondFile.new(
1751         part,
1752         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1753         )
1754 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
1755 pdf_path = f'{directory}/Section_C.pdf'
1756 path = pathlib.Path('Section_C.pdf')
1757 if path.exists():
1758     print(f'Removing {pdf_path} ...')
1759     path.unlink()
1760 time_1 = time.time()
1761 print(f'Persisting {pdf_path} ...')
1762 result = abjad.persist(part_file).as_pdf(pdf_path)
1763 print(result[0])
1764 print(result[1])
1765 print(result[2])
1766 success = result[3]
1767 if success is False:
1768     print('LilyPond failed!')
1769 time_2 = time.time()
1770 total_time = time_2 - time_1
1771 print(f'Total time: {total_time} seconds')
1772 if path.exists():
1773     print(f'Opening {pdf_path} ...')
1774     os.system(f'open {pdf_path}')

```

```

1775 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/
1776 Section_C.ly').readlines()
1777 open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/Section_C.ly',
1778 'w').writelines(part_lines[15:-1])
1779
1780 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
1781     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1782     signature_copy = abjad.mutate(signatures).copy()
1783     staff_copy = abjad.mutate(staff).copy()
1784     part = abjad.Score()
1785     part.insert(0, staff)
1786     part.insert(0, signature_copy)
1787     part_file = abjad.LilyPondFile.new(
1788         part,
1789         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1790 _stylesheets/abjad.ily'],
1791         )
1792     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3'
1793     pdf_path = f'{directory}/Section_C.pdf'
1794     path = pathlib.Path('Section_C.pdf')
1795     if path.exists():
1796         print(f'Removing {pdf_path} ...')
1797         path.unlink()
1798     time_1 = time.time()
1799     print(f'Persisting {pdf_path} ...')
1800     result = abjad.persist(part_file).as_pdf(pdf_path)
1801     print(result[0])
1802     print(result[1])
1803     print(result[2])
1804     success = result[3]
1805     if success is False:
1806         print('LilyPond failed!')
1807     time_2 = time.time()
1808     total_time = time_2 - time_1
1809     print(f'Total time: {total_time} seconds')
1810     if path.exists():
1811         print(f'Opening {pdf_path} ...')
1812         os.system(f'open {pdf_path}')
1813     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/
1814 Section_C.ly').readlines()
1815     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/Section_C.ly',
1816 'w').writelines(part_lines[15:-1])
1817
1818 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
1819     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1820     signature_copy = abjad.mutate(signatures).copy()
1821     staff_copy = abjad.mutate(staff).copy()
1822     part = abjad.Score()
1823     part.insert(0, staff)
1824     part.insert(0, signature_copy)
1825     part_file = abjad.LilyPondFile.new(
1826         part,
1827         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1828 _stylesheets/abjad.ily'],
1829         )
1830     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1'
1831     pdf_path = f'{directory}/Section_C.pdf'
1832     path = pathlib.Path('Section_C.pdf')
1833     if path.exists():

```

```

1828     print(f'Removing {pdf_path} ...')
1829     path.unlink()
1830 time_1 = time.time()
1831 print(f'Persisting {pdf_path} ...')
1832 result = abjad.persist(part_file).as_pdf(pdf_path)
1833 print(result[0])
1834 print(result[1])
1835 print(result[2])
1836 success = result[3]
1837 if success is False:
1838     print('LilyPond failed!')
1839 time_2 = time.time()
1840 total_time = time_2 - time_1
1841 print(f'Total time: {total_time} seconds')
1842 if path.exists():
1843     print(f'Opening {pdf_path} ...')
1844     os.system(f'open {pdf_path}')
1845 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/
Section_C.ly').readlines()
1846 open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/Section_C.ly', 'w'
).writelines(part_lines[15:-1])
1847
1848 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
1849 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1850 signature_copy = abjad.mutate(signatures).copy()
1851 staff_copy = abjad.mutate(staff).copy()
1852 part = abjad.Score()
1853 part.insert(0, staff)
1854 part.insert(0, signature_copy)
1855 part_file = abjad.LilyPondFile.new(
1856     part,
1857     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1858     )
1859 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
1860 pdf_path = f'{directory}/Section_C.pdf'
1861 path = pathlib.Path('Section_C.pdf')
1862 if path.exists():
1863     print(f'Removing {pdf_path} ...')
1864     path.unlink()
1865 time_1 = time.time()
1866 print(f'Persisting {pdf_path} ...')
1867 result = abjad.persist(part_file).as_pdf(pdf_path)
1868 print(result[0])
1869 print(result[1])
1870 print(result[2])
1871 success = result[3]
1872 if success is False:
1873     print('LilyPond failed!')
1874 time_2 = time.time()
1875 total_time = time_2 - time_1
1876 print(f'Total time: {total_time} seconds')
1877 if path.exists():
1878     print(f'Opening {pdf_path} ...')
1879     os.system(f'open {pdf_path}')
1880 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
Section_C.ly').readlines()
1881 open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_C.ly', 'w'
).writelines(part_lines[15:-1])

```

```

1882
1883 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
1884     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1885     signature_copy = abjad.mutate(signatures).copy()
1886     staff_copy = abjad.mutate(staff).copy()
1887     part = abjad.Score()
1888     part.insert(0, staff)
1889     part.insert(0, signature_copy)
1890     part_file = abjad.LilyPondFile.new(
1891         part,
1892         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1893         _stylesheets/abjad.ily'],
1894         )
1895     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
1896     pdf_path = f'{directory}/Section_C.pdf'
1897     path = pathlib.Path('Section_C.pdf')
1898     if path.exists():
1899         print(f'Removing {pdf_path} ...')
1900         path.unlink()
1901     time_1 = time.time()
1902     print(f'Persisting {pdf_path} ...')
1903     result = abjad.persist(part_file).as_pdf(pdf_path)
1904     print(result[0])
1905     print(result[1])
1906     print(result[2])
1907     success = result[3]
1908     if success is False:
1909         print('LilyPond failed!')
1910     time_2 = time.time()
1911     total_time = time_2 - time_1
1912     print(f'Total time: {total_time} seconds')
1913     if path.exists():
1914         print(f'Opening {pdf_path} ...')
1915         os.system(f'open {pdf_path}')
1916     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/
1917     Section_C.ly').readlines()
1918     open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/Section_C.ly', 'w'
1919     ).writelines(part_lines[15:-1])

1920
1921 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
1922     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1923     signature_copy = abjad.mutate(signatures).copy()
1924     staff_copy = abjad.mutate(staff).copy()
1925     part = abjad.Score()
1926     part.insert(0, staff)
1927     part.insert(0, signature_copy)
1928     part_file = abjad.LilyPondFile.new(
1929         part,
1930         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1931         _stylesheets/abjad.ily'],
1932         )
1933     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
1934     pdf_path = f'{directory}/Section_C.pdf'
1935     path = pathlib.Path('Section_C.pdf')
1936     if path.exists():
1937         print(f'Removing {pdf_path} ...')
1938         path.unlink()
1939     time_1 = time.time()
1940     print(f'Persisting {pdf_path} ...')

```

```

1937     result = abjad.persist(part_file).as_pdf(pdf_path)
1938     print(result[0])
1939     print(result[1])
1940     print(result[2])
1941     success = result[3]
1942     if success is False:
1943         print('LilyPond failed!')
1944     time_2 = time.time()
1945     total_time = time_2 - time_1
1946     print(f'Total time: {total_time} seconds')
1947     if path.exists():
1948         print(f'Opening {pdf_path} ...')
1949         os.system(f'open {pdf_path}')
1950     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/
Section_C.ly').readlines()
1951     open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/Section_C.ly', 'w'
).writelines(part_lines[15:-1])
1952
1953 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
1954     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1955     signature_copy = abjad.mutate(signatures).copy()
1956     staff_copy = abjad.mutate(staff).copy()
1957     part = abjad.Score()
1958     part.insert(0, staff)
1959     part.insert(0, signature_copy)
1960     part_file = abjad.LilyPondFile.new(
1961         part,
1962         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1963         )
1964     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5'
1965     pdf_path = f'{directory}/Section_C.pdf'
1966     path = pathlib.Path('Section_C.pdf')
1967     if path.exists():
1968         print(f'Removing {pdf_path} ...')
1969         path.unlink()
1970     time_1 = time.time()
1971     print(f'Persisting {pdf_path} ...')
1972     result = abjad.persist(part_file).as_pdf(pdf_path)
1973     print(result[0])
1974     print(result[1])
1975     print(result[2])
1976     success = result[3]
1977     if success is False:
1978         print('LilyPond failed!')
1979     time_2 = time.time()
1980     total_time = time_2 - time_1
1981     print(f'Total time: {total_time} seconds')
1982     if path.exists():
1983         print(f'Opening {pdf_path} ...')
1984         os.system(f'open {pdf_path}')
1985     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/
Section_C.ly').readlines()
1986     open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/Section_C.ly', 'w'
).writelines(part_lines[15:-1])
1987
1988 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
1989     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1990     signature_copy = abjad.mutate(signatures).copy()

```

```

1991 staff_copy = abjad.mutate(staff).copy()
1992 part = abjad.Score()
1993 part.insert(0, staff)
1994 part.insert(0, signature_copy)
1995 part_file = abjad.LilyPondFile.new(
1996     part,
1997     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1998 _stylesheets/abjad.ily'],
1999 )
2000 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
2001 pdf_path = f'{directory}/Section_C.pdf'
2002 path = pathlib.Path('Section_C.pdf')
2003 if path.exists():
2004     print(f'Removing {pdf_path} ...')
2005     path.unlink()
2006 time_1 = time.time()
2007 print(f'Persisting {pdf_path} ...')
2008 result = abjad.persist(part_file).as_pdf(pdf_path)
2009 print(result[0])
2010 print(result[1])
2011 print(result[2])
2012 success = result[3]
2013 if success is False:
2014     print('LilyPond failed!')
2015 time_2 = time.time()
2016 total_time = time_2 - time_1
2017 print(f'Total time: {total_time} seconds')
2018 if path.exists():
2019     print(f'Opening {pdf_path} ...')
2020     os.system(f'open {pdf_path}')
2021 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
2022 Section_C.ly').readlines()
2023 open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_C.ly', 'w'
2024 .writelines(part_lines[15:-1])
2025
2026 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
2027     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2028     signature_copy = abjad.mutate(signatures).copy()
2029     staff_copy = abjad.mutate(staff).copy()
2030     part = abjad.Score()
2031     part.insert(0, staff)
2032     part.insert(0, signature_copy)
2033     part_file = abjad.LilyPondFile.new(
2034         part,
2035         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2036 _stylesheets/abjad.ily'],
2037     )
2038     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
2039     pdf_path = f'{directory}/Section_C.pdf'
2040     path = pathlib.Path('Section_C.pdf')
2041     if path.exists():
2042         print(f'Removing {pdf_path} ...')
2043         path.unlink()
2044     time_1 = time.time()
2045     print(f'Persisting {pdf_path} ...')
2046     result = abjad.persist(part_file).as_pdf(pdf_path)
2047     print(result[0])
2048     print(result[1])
2049     print(result[2])

```

```

2046     success = result[3]
2047     if success is False:
2048         print('LilyPond failed!')
2049     time_2 = time.time()
2050     total_time = time_2 - time_1
2051     print(f'Total time: {total_time} seconds')
2052     if path.exists():
2053         print(f'Opening {pdf_path} ...')
2054         os.system(f'open {pdf_path}')
2055     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.')tenor1/
2056     Section_C.ly').readlines()
2057     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.')tenor1/Section_C.ly', ,
2058     w').writelines(part_lines[15:-1])
2059
2060 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
2061     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2062     signature_copy = abjad.mutate(signatures).copy()
2063     staff_copy = abjad.mutate(staff).copy()
2064     part = abjad.Score()
2065     part.insert(0, staff)
2066     part.insert(0, signature_copy)
2067     part_file = abjad.LilyPondFile.new(
2068         part,
2069         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2070         _stylesheets/abjad.ily'],
2071         )
2072     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2'
2073     pdf_path = f'{directory}/Section_C.pdf'
2074     path = pathlib.Path('Section_C.pdf')
2075     if path.exists():
2076         print(f'Removing {pdf_path} ...')
2077         path.unlink()
2078     time_1 = time.time()
2079     print(f'Persisting {pdf_path} ...')
2080     result = abjad.persist(part_file).as_pdf(pdf_path)
2081     print(result[0])
2082     print(result[1])
2083     print(result[2])
2084     success = result[3]
2085     if success is False:
2086         print('LilyPond failed!')
2087     time_2 = time.time()
2088     total_time = time_2 - time_1
2089     print(f'Total time: {total_time} seconds')
2090     if path.exists():
2091         print(f'Opening {pdf_path} ...')
2092         os.system(f'open {pdf_path}')
2093     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/
2094     Section_C.ly').readlines()
2095     open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/Section_C.ly', ,
2096     w').writelines(part_lines[15:-1])
2097
2098 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
2099     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2100     signature_copy = abjad.mutate(signatures).copy()
2101     staff_copy = abjad.mutate(staff).copy()
2102     part = abjad.Score()
2103     part.insert(0, staff)
2104     part.insert(0, signature_copy)

```

```

2100 part_file = abjad.LilyPondFile.new(
2101     part,
2102     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2103 _stylesheets/abjad.ily'],
2104     )
2105 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3'
2106 pdf_path = f'{directory}/Section_C.pdf'
2107 path = pathlib.Path('Section_C.pdf')
2108 if path.exists():
2109     print(f'Removing {pdf_path} ...')
2110     path.unlink()
2111 time_1 = time.time()
2112 print(f'Persisting {pdf_path} ...')
2113 result = abjad.persist(part_file).as_pdf(pdf_path)
2114 print(result[0])
2115 print(result[1])
2116 print(result[2])
2117 success = result[3]
2118 if success is False:
2119     print('LilyPond failed!')
2120 time_2 = time.time()
2121 total_time = time_2 - time_1
2122 print(f'Total time: {total_time} seconds')
2123 if path.exists():
2124     print(f'Opening {pdf_path} ...')
2125     os.system(f'open {pdf_path}')
2126 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
2127 Section_C.ly').readlines()
2128 open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_C.ly', 'w'
2129 .writelines(part_lines[15:-1])

for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
2130     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2131     signature_copy = abjad.mutate(signatures).copy()
2132     staff_copy = abjad.mutate(staff).copy()
2133     part = abjad.Score()
2134     part.insert(0, staff)
2135     part.insert(0, signature_copy)
2136     part_file = abjad.LilyPondFile.new(
2137         part,
2138         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2139 _stylesheets/abjad.ily'],
2140         )
2141 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'
2142 pdf_path = f'{directory}/Section_C.pdf'
2143 path = pathlib.Path('Section_C.pdf')
2144 if path.exists():
2145     print(f'Removing {pdf_path} ...')
2146     path.unlink()
2147 time_1 = time.time()
2148 print(f'Persisting {pdf_path} ...')
2149 result = abjad.persist(part_file).as_pdf(pdf_path)
2150 print(result[0])
2151 print(result[1])
2152 print(result[2])
2153 success = result[3]
2154 if success is False:
2155     print('LilyPond failed!')
time_2 = time.time()

```

```

2155     total_time = time_2 - time_1
2156     print(f'Total time: {total_time} seconds')
2157     if path.exists():
2158         print(f'Opening {pdf_path} ...')
2159         os.system(f'open {pdf_path}')
2160     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
2161     Section_C.ly').readlines()
2162     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_C.ly', ,
2163     w').writelines(part_lines[15:-1])
2164
2165 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
2166     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2167     signature_copy = abjad.mutate(signatures).copy()
2168     staff_copy = abjad.mutate(staff).copy()
2169     part = abjad.Score()
2170     part.insert(0, staff)
2171     part.insert(0, signature_copy)
2172     part_file = abjad.LilyPondFile.new(
2173         part,
2174         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2175         _stylesheets/abjad.ily'],
2176         )
2177     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5'
2178     pdf_path = f'{directory}/Section_C.pdf'
2179     path = pathlib.Path('Section_C.pdf')
2180     if path.exists():
2181         print(f'Removing {pdf_path} ...')
2182         path.unlink()
2183     time_1 = time.time()
2184     print(f'Persisting {pdf_path} ...')
2185     result = abjad.persist(part_file).as_pdf(pdf_path)
2186     print(result[0])
2187     print(result[1])
2188     print(result[2])
2189     success = result[3]
2190     if success is False:
2191         print('LilyPond failed!')
2192     time_2 = time.time()
2193     total_time = time_2 - time_1
2194     print(f'Total time: {total_time} seconds')
2195     if path.exists():
2196         print(f'Opening {pdf_path} ...')
2197         os.system(f'open {pdf_path}')
2198     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/
2199     Section_C.ly').readlines()
2200     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/Section_C.ly', ,
2201     w').writelines(part_lines[15:-1])
2202
2203 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
2204     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2205     signature_copy = abjad.mutate(signatures).copy()
2206     staff_copy = abjad.mutate(staff).copy()
2207     part = abjad.Score()
2208     part.insert(0, staff)
2209     part.insert(0, signature_copy)
2210     part_file = abjad.LilyPondFile.new(
2211         part,
2212         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2213         _stylesheets/abjad.ily'],

```

```

2208     )
2209     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1'
2210     pdf_path = f'{directory}/Section_C.pdf'
2211     path = pathlib.Path('Section_C.pdf')
2212     if path.exists():
2213         print(f'Removing {pdf_path} ...')
2214         path.unlink()
2215     time_1 = time.time()
2216     print(f'Persisting {pdf_path} ...')
2217     result = abjad.persist(part_file).as_pdf(pdf_path)
2218     print(result[0])
2219     print(result[1])
2220     print(result[2])
2221     success = result[3]
2222     if success is False:
2223         print('LilyPond failed!')
2224     time_2 = time.time()
2225     total_time = time_2 - time_1
2226     print(f'Total time: {total_time} seconds')
2227     if path.exists():
2228         print(f'Opening {pdf_path} ...')
2229         os.system(f'open {pdf_path}')
2230     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/
2231     Section_C.ly').readlines()
2232     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/Section_C.ly',
2233     'w').writelines(part_lines[15:-1])
2234
2235 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
2236     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2237     signature_copy = abjad.mutate(signatures).copy()
2238     staff_copy = abjad.mutate(staff).copy()
2239     part = abjad.Score()
2240     part.insert(0, staff)
2241     part.insert(0, signature_copy)
2242     part_file = abjad.LilyPondFile.new(
2243         part,
2244         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2245         _stylesheets/abjad.ily'],
2246         )
2247     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2'
2248     pdf_path = f'{directory}/Section_C.pdf'
2249     path = pathlib.Path('Section_C.pdf')
2250     if path.exists():
2251         print(f'Removing {pdf_path} ...')
2252         path.unlink()
2253     time_1 = time.time()
2254     print(f'Persisting {pdf_path} ...')
2255     result = abjad.persist(part_file).as_pdf(pdf_path)
2256     print(result[0])
2257     print(result[1])
2258     print(result[2])
2259     success = result[3]
2260     if success is False:
2261         print('LilyPond failed!')
2262     time_2 = time.time()
2263     total_time = time_2 - time_1
2264     print(f'Total time: {total_time} seconds')
2265     if path.exists():
2266         print(f'Opening {pdf_path} ...')

```

```

2264     os.system(f'open {pdf_path}')
2265     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/
2266     Section_C.ly').readlines()
2267     open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/Section_C.ly',
2268     , 'w').writelines(part_lines[15:-1])
2269
2270 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
2271     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2272     signature_copy = abjad.mutate(signatures).copy()
2273     staff_copy = abjad.mutate(staff).copy()
2274     part = abjad.Score()
2275     part.insert(0, staff)
2276     part.insert(0, signature_copy)
2277     part_file = abjad.LilyPondFile.new(
2278         part,
2279         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2280         _stylesheets/abjad.ily'],
2281         )
2282     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3'
2283     pdf_path = f'{directory}/Section_C.pdf'
2284     path = pathlib.Path('Section_C.pdf')
2285     if path.exists():
2286         print(f'Removing {pdf_path} ...')
2287         path.unlink()
2288     time_1 = time.time()
2289     print(f'Persisting {pdf_path} ...')
2290     result = abjad.persist(part_file).as_pdf(pdf_path)
2291     print(result[0])
2292     print(result[1])
2293     print(result[2])
2294     success = result[3]
2295     if success is False:
2296         print('LilyPond failed!')
2297     time_2 = time.time()
2298     total_time = time_2 - time_1
2299     print(f'Total time: {total_time} seconds')
2300     if path.exists():
2301         print(f'Opening {pdf_path} ...')
2302         os.system(f'open {pdf_path}')
2303     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/
2304     Section_C.ly').readlines()
2305     open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/Section_C.ly',
2306     , 'w').writelines(part_lines[15:-1])
2307
2308 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
2309     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2310     signature_copy = abjad.mutate(signatures).copy()
2311     staff_copy = abjad.mutate(staff).copy()
2312     part = abjad.Score()
2313     part.insert(0, staff)
2314     part.insert(0, signature_copy)
2315     part_file = abjad.LilyPondFile.new(
2316         part,
2317         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2318         _stylesheets/abjad.ily'],
2319         )
2320     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.')bass1'
2321     pdf_path = f'{directory}/Section_C.pdf'
2322     path = pathlib.Path('Section_C.pdf')

```

```

2317     if path.exists():
2318         print(f'Removing {pdf_path} ...')
2319         path.unlink()
2320     time_1 = time.time()
2321     print(f'Persisting {pdf_path} ...')
2322     result = abjad.persist(part_file).as_pdf(pdf_path)
2323     print(result[0])
2324     print(result[1])
2325     print(result[2])
2326     success = result[3]
2327     if success is False:
2328         print('LilyPond failed!')
2329     time_2 = time.time()
2330     total_time = time_2 - time_1
2331     print(f'Total time: {total_time} seconds')
2332     if path.exists():
2333         print(f'Opening {pdf_path} ...')
2334         os.system(f'open {pdf_path}')
2335     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
2336     Section_C.ly').readlines()
2337     open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_C.ly', 'w'
2338     ).writelines(part_lines[15:-1])
2339
2340 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
2341     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2342     signature_copy = abjad.mutate(signatures).copy()
2343     staff_copy = abjad.mutate(staff).copy()
2344     part = abjad.Score()
2345     part.insert(0, staff)
2346     part.insert(0, signature_copy)
2347     part_file = abjad.LilyPondFile.new(
2348         part,
2349         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2350         _stylesheets/abjad.ily'],
2351         )
2352     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
2353     pdf_path = f'{directory}/Section_C.pdf'
2354     path = pathlib.Path('Section_C.pdf')
2355     if path.exists():
2356         print(f'Removing {pdf_path} ...')
2357         path.unlink()
2358     time_1 = time.time()
2359     print(f'Persisting {pdf_path} ...')
2360     result = abjad.persist(part_file).as_pdf(pdf_path)
2361     print(result[0])
2362     print(result[1])
2363     print(result[2])
2364     success = result[3]
2365     if success is False:
2366         print('LilyPond failed!')
2367     time_2 = time.time()
2368     total_time = time_2 - time_1
2369     print(f'Total time: {total_time} seconds')
2370     if path.exists():
2371         print(f'Opening {pdf_path} ...')
2372         os.system(f'open {pdf_path}')
2373     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
2374     Section_C.ly').readlines()
2375     open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_C.ly', 'w'

```

```

2372
2373 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
2374     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2375     signature_copy = abjad.mutate(signatures).copy()
2376     staff_copy = abjad.mutate(staff).copy()
2377     part = abjad.Score()
2378     part.insert(0, staff)
2379     part.insert(0, signature_copy)
2380     part_file = abjad.LilyPondFile.new(
2381         part,
2382         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2383         _stylesheets/abjad.ily'],
2384         )
2385     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
2386     pdf_path = f'{directory}/Section_C.pdf'
2387     path = pathlib.Path('Section_C.pdf')
2388     if path.exists():
2389         print(f'Removing {pdf_path} ...')
2390         path.unlink()
2391     time_1 = time.time()
2392     print(f'Persisting {pdf_path} ...')
2393     result = abjad.persist(part_file).as_pdf(pdf_path)
2394     print(result[0])
2395     print(result[1])
2396     print(result[2])
2397     success = result[3]
2398     if success is False:
2399         print('LilyPond failed!')
2400     time_2 = time.time()
2401     total_time = time_2 - time_1
2402     print(f'Total time: {total_time} seconds')
2403     if path.exists():
2404         print(f'Opening {pdf_path} ...')
2405         os.system(f'open {pdf_path}')
2406     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass
2407     /Section_C.ly').readlines()
2408     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_C.ly
2409     ', 'w').writelines(part_lines[15:-1])

```

Listing 3.4: Invocation Source Code

## 3.5 Section D

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11
12 print('Interpreting file ...')
13

```

```

14 time_signatures = [
15     abjad.TimeSignature(pair) for pair in [
16         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
17         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
18         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
19     ]
20 ]
21
22 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
23
24 def cyc(lst):
25     count = 0
26     while True:
27         yield lst[count%len(lst)]
28         count += 1
29
30 def grouper(lst1, lst2):
31     def cyc(lst):
32         c = 0
33         while True:
34             yield lst[c%len(lst)]
35             c += 1
36     lst1 = cyc(lst1)
37     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2
38 ]
39
40 def reduceMod(list_length, rw):
41     return [(x % list_length) for x in rw]
42
43 # -3 at bottom of chord for completion
44 soprano_chord = [27, ]
45 soprano_1_chord = [13.25, 16, 22,]
46 soprano_2_chord = [13, 14.75, 16,]
47 soprano_3_chord = [12.75, 15.5, 13,]
48 alto_1_chord = [12.5, 19, 20,]
49 alto_2_chord = [12.5, 15.25, 25.5, 12,]
50 alto_3_chord = [1.75, 13.5, 22.25, 1,]
51 alto_4_chord = [12.5, 15.25, 25.5, 20,]
52 alto_5_chord = [1.75, 13.5, 22.25, 12,]
53 alto_6_chord = [12.5, 19, 1,]
54 tenor_1_chord = [6, 17.5, 17,]
55 tenor_2_chord = [6, 17.5, 25.5, 6,]
56 tenor_3_chord = [6, 17.5, 25.5, -1]
57 tenor_4_chord = [6, 17.5, 17,]
58 tenor_5_chord = [6, 17.5, 25.5, 6,]
59 baritone_1_chord = [13.25, 13,]
60 baritone_2_chord = [4, 16.5, 23.5, 6,]
61 baritone_3_chord = [7.75, 17.75, 25.5, 4,]
62 bass_1_chord = [11, 9, 9, 11, 9, 11,]
63 bass_2_chord = [9, 11, 11, 9, 11, 9,]
64 contrabass_chord = [-2, 2, 7, -2, 2, 7, 2, -2]
65
66 def reduceMod(x, rw):
67     return [(y % x) for y in rw]
68
69 seed(43)
70 soprano_random_walk = []
71 soprano_random_walk.append(-1 if random() < 0.5 else 1)
72 for i in range(1, 1000):

```

```

72     movement = -1 if random() < 0.5 else 1
73     value = sopranino_random_walk[i-1] + movement
74     sopranino_random_walk.append(value)
75     sopranino_walk_chord = [18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14,
76     13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5, 6,
77     7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ]
78 l = len(sopranino_walk_chord)
79 sopranino_random_walk_notes = [sopranino_walk_chord[x] for x in reduceMod(l,
80                             sopranino_random_walk)]
81
82
83 seed(44)
84 soprano_1_random_walk = []
85 soprano_1_random_walk.append(-1 if random() < 0.5 else 1)
86 for i in range(1, 1000):
87     movement = -1 if random() < 0.5 else 1
88     value = soprano_1_random_walk[i-1] + movement
89     soprano_1_random_walk.append(value)
90 soprano_1_walk_chord = [17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14,
91     13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5, 6,
92     7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ]
93 l = len(soprano_1_walk_chord)
94 soprano_1_random_walk_notes = [soprano_1_walk_chord[x] for x in reduceMod(l,
95                             soprano_1_random_walk)]
96
97 seed(45)
98 soprano_2_random_walk = []
99 soprano_2_random_walk.append(-1 if random() < 0.5 else 1)
100 for i in range(1, 1000):
101     movement = -1 if random() < 0.5 else 1
102     value = soprano_2_random_walk[i-1] + movement
103     soprano_2_random_walk.append(value)
104 soprano_2_walk_chord = [16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15,
105     14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5,
106     6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ]
107 l = len(soprano_2_walk_chord)
108 soprano_2_random_walk_notes = [soprano_2_walk_chord[x] for x in reduceMod(l,
109                             soprano_2_random_walk)]
110
111 seed(46)
112 soprano_3_random_walk = []
113 soprano_3_random_walk.append(-1 if random() < 0.5 else 1)
114 for i in range(1, 1000):
115     movement = -1 if random() < 0.5 else 1
116     value = soprano_3_random_walk[i-1] + movement
117     soprano_3_random_walk.append(value)
118 soprano_3_walk_chord = [15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16,
119     15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3,
120     4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ]
121 l = len(soprano_3_walk_chord)
122 soprano_3_random_walk_notes = [soprano_3_walk_chord[x] for x in reduceMod(l,
123                             soprano_3_random_walk)]
124
125 seed(47)
126 alto_1_random_walk = []
127 alto_1_random_walk.append(-1 if random() < 0.5 else 1)
128 for i in range(1, 1000):
129     movement = -1 if random() < 0.5 else 1

```

```

119     value = alto_1_random_walk[i-1] + movement
120     alto_1_random_walk.append(value)
121 alto_1_walk_chord = [14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16,
122     15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3,
123     4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ]
124 l = len(alto_1_walk_chord)
125 alto_1_random_walk_notes = [alto_1_walk_chord[x] for x in reduceMod(l,
126     alto_1_random_walk)]
127
128 seed(48)
129 alto_2_random_walk = []
130 alto_2_random_walk.append(-1 if random() < 0.5 else 1)
131 for i in range(1, 1000):
132     movement = -1 if random() < 0.5 else 1
133     value = alto_2_random_walk[i-1] + movement
134     alto_2_random_walk.append(value)
135 alto_2_walk_chord = [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17,
136     16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2,
137     3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ]
138 l = len(alto_2_walk_chord)
139 alto_2_random_walk_notes = [alto_2_walk_chord[x] for x in reduceMod(l,
140     alto_2_random_walk)]
141
142 seed(49)
143 alto_3_random_walk = []
144 alto_3_random_walk.append(-1 if random() < 0.5 else 1)
145 for i in range(1, 1000):
146     movement = -1 if random() < 0.5 else 1
147     value = alto_3_random_walk[i-1] + movement
148     alto_3_random_walk.append(value)
149 alto_3_walk_chord = [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18,
150     17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1,
151     2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ]
152 l = len(alto_3_walk_chord)
153 alto_3_random_walk_notes = [alto_3_walk_chord[x] for x in reduceMod(l,
154     alto_3_random_walk)]
155
156 seed(50)
157 alto_4_random_walk = []
158 alto_4_random_walk.append(-1 if random() < 0.5 else 1)
159 for i in range(1, 1000):
160     movement = -1 if random() < 0.5 else 1
161     value = alto_4_random_walk[i-1] + movement
162     alto_4_random_walk.append(value)
163 alto_4_walk_chord = [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19,
164     18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1,
165     0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ]
166 l = len(alto_4_walk_chord)
167 alto_4_random_walk_notes = [alto_4_walk_chord[x] for x in reduceMod(l,
168     alto_4_random_walk)]
169
170 seed(51)
171 alto_5_random_walk = []
172 alto_5_random_walk.append(-1 if random() < 0.5 else 1)
173 for i in range(1, 1000):
174     movement = -1 if random() < 0.5 else 1
175     value = alto_5_random_walk[i-1] + movement
176     alto_5_random_walk.append(value)
177 alto_5_walk_chord = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20,
```

```

    19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2,
    -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ]
166 l = len(alto_5_walk_chord)
167 alto_5_random_walk_notes = [alto_5_walk_chord[x] for x in reduceMod(l,
    alto_5_random_walk)]
168
169 seed(52)
170 alto_6_random_walk = []
171 alto_6_random_walk.append(-1 if random() < 0.5 else 1)
172 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = alto_6_random_walk[i-1] + movement
    alto_6_random_walk.append(value)
173 alto_6_walk_chord = [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21,
    20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1,
    -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, ]
177 l = len(alto_6_walk_chord)
178 alto_6_random_walk_notes = [alto_6_walk_chord[x] for x in reduceMod(l,
    alto_6_random_walk)]
179
180 seed(53)
181 tenor_1_random_walk = []
182 tenor_1_random_walk.append(-1 if random() < 0.5 else 1)
183 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = tenor_1_random_walk[i-1] + movement
    tenor_1_random_walk.append(value)
184 tenor_1_walk_chord = [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
    -1, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, ]
188 l = len(tenor_1_walk_chord)
189 tenor_1_random_walk_notes = [tenor_1_walk_chord[x] for x in reduceMod(l,
    tenor_1_random_walk)]
190
191 seed(54)
192 tenor_2_random_walk = []
193 tenor_2_random_walk.append(-1 if random() < 0.5 else 1)
194 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = tenor_2_random_walk[i-1] + movement
    tenor_2_random_walk.append(value)
195 tenor_2_walk_chord = [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
    -1, -2, -1, 0, 1, 2, 3, 4, 5, 6, ]
199 l = len(tenor_2_walk_chord)
200 tenor_2_random_walk_notes = [tenor_2_walk_chord[x] for x in reduceMod(l,
    tenor_2_random_walk)]
201
202 seed(55)
203 tenor_3_random_walk = []
204 tenor_3_random_walk.append(-1 if random() < 0.5 else 1)
205 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = tenor_3_random_walk[i-1] + movement
    tenor_3_random_walk.append(value)
206 tenor_3_walk_chord = [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
    22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1,
    0, -1, -2, -1, 0, 1, 2, 3, 4, 5, ]
210 l = len(tenor_3_walk_chord)

```

```

211 tenor_3_random_walk_notes = [tenor_3_walk_chord[x] for x in reduceMod(1,
212   tenor_3_random_walk)]
213
214 seed(56)
215 tenor_4_random_walk = []
216 tenor_4_random_walk.append(-1 if random() < 0.5 else 1)
217 for i in range(1, 1000):
218     movement = -1 if random() < 0.5 else 1
219     value = tenor_4_random_walk[i-1] + movement
220     tenor_4_random_walk.append(value)
221 tenor_4_walk_chord = [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
222   21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2,
223   1, 0, -1, -2, -1, 0, 1, 2, 3, 4, ]
224 l = len(tenor_4_walk_chord)
225 tenor_4_random_walk_notes = [tenor_4_walk_chord[x] for x in reduceMod(1,
226   tenor_4_random_walk)]
227
228 seed(57)
229 tenor_5_random_walk = []
230 tenor_5_random_walk.append(-1 if random() < 0.5 else 1)
231 for i in range(1, 1000):
232     movement = -1 if random() < 0.5 else 1
233     value = tenor_5_random_walk[i-1] + movement
234     tenor_5_random_walk.append(value)
235 tenor_5_walk_chord = [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
236   20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4,
237   3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, ]
238 l = len(tenor_5_walk_chord)
239 tenor_5_random_walk_notes = [tenor_5_walk_chord[x] for x in reduceMod(1,
240   tenor_5_random_walk)]
241
242 seed(58)
243 baritone_1_random_walk = []
244 baritone_1_random_walk.append(-1 if random() < 0.5 else 1)
245 for i in range(1, 1000):
246     movement = -1 if random() < 0.5 else 1
247     value = baritone_1_random_walk[i-1] + movement
248     baritone_1_random_walk.append(value)
249 baritone_1_walk_chord = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
250   19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5,
251   4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, ]
252 l = len(baritone_1_walk_chord)
253 baritone_1_random_walk_notes = [baritone_1_walk_chord[x] for x in reduceMod(1,
254   baritone_1_random_walk)]
255
256 seed(59)
257 baritone_2_random_walk = []
258 baritone_2_random_walk.append(-1 if random() < 0.5 else 1)
259 for i in range(1, 1000):
260     movement = -1 if random() < 0.5 else 1
261     value = baritone_2_random_walk[i-1] + movement
262     baritone_2_random_walk.append(value)
263 baritone_2_walk_chord = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
264   18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,
265   5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, ]
266 l = len(baritone_2_walk_chord)
267 baritone_2_random_walk_notes = [baritone_2_walk_chord[x] for x in reduceMod(1,
268   baritone_2_random_walk)]

```

```

257 seed(60)
258 baritone_3_random_walk = []
259 baritone_3_random_walk.append(-1 if random() < 0.5 else 1)
260 for i in range(1, 1000):
261     movement = -1 if random() < 0.5 else 1
262     value = baritone_3_random_walk[i-1] + movement
263     baritone_3_random_walk.append(value)
264 baritone_3_walk_chord = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
265     17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8,
266     7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, ]
267 l = len(baritone_3_walk_chord)
268 baritone_3_random_walk_notes = [baritone_3_walk_chord[x] for x in reduceMod(l,
269     baritone_3_random_walk)]
270
271 seed(61)
272 bass_1_random_walk = []
273 bass_1_random_walk.append(-1 if random() < 0.5 else 1)
274 for i in range(1, 1000):
275     movement = -1 if random() < 0.5 else 1
276     value = bass_1_random_walk[i-1] + movement
277     bass_1_random_walk.append(value)
278 bass_1_walk_chord = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
279     18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7,
280     6, 5, 4, 3, 2, 1, 0, -1, -2, -1, ]
281 l = len(bass_1_walk_chord)
282 bass_1_random_walk_notes = [bass_1_walk_chord[x] for x in reduceMod(l,
283     bass_1_random_walk)]
284
285 seed(62)
286 bass_2_random_walk = []
287 bass_2_random_walk.append(-1 if random() < 0.5 else 1)
288 for i in range(1, 1000):
289     movement = -1 if random() < 0.5 else 1
290     value = bass_2_random_walk[i-1] + movement
291     bass_2_random_walk.append(value)
292 bass_2_walk_chord = [-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
293     17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8,
294     7, 6, 5, 4, 3, 2, 1, 0, -1, -2, ]
295 l = len(bass_2_walk_chord)
296 bass_2_random_walk_notes = [bass_2_walk_chord[x] for x in reduceMod(l,
297     bass_2_random_walk)]
298
299 seed(63)
300 contrabass_random_walk = []
301 contrabass_random_walk.append(-1 if random() < 0.5 else 1)
302 for i in range(1, 1000):
303     movement = -1 if random() < 0.5 else 1
304     value = contrabass_random_walk[i-1] + movement
305     contrabass_random_walk.append(value)
306 contrabass_walk_chord = [-2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
307     15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11,
308     10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, ]
309 l = len(contrabass_walk_chord)
310 contrabass_random_walk_notes = [contrabass_walk_chord[x] for x in reduceMod(l,
311     contrabass_random_walk)]
312
313 rmaker_two = abjadext.rmakers.NoteRhythmMaker()
314
315 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(

```

```

304     talea=abjadext.rmakers.Talea(
305         counts=[1, 1, 1, 3, 2, 1, 2, 4, 1, 3, 2, 3, 2, 1, 4, 1, 5, 2, 1, 3, ],
306         denominator=16,
307         ),
308     beamSpecifier=abjadext.rmakers.BeamSpecifier(
309         beamDivisionsTogether=True,
310         beamRests=False,
311         ),
312     extraCountsPerDivision=[0, 1, -1, 1, 0, -1, 0, ],
313     logicalTieMasks=[
314         abjadext.rmakers.silence([6], 11),
315         ],
316     divisionMasks=[
317         abjadext.rmakers.SilenceMask(
318             pattern=abjad.index([6], 15),
319             ),
320         ],
321     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
322         trivialize=True,
323         extractTrivial=True,
324         rewriteRestFilled=True,
325         rewriteSustained=True,
326         ),
327     )
328
329 # Initialize AttachmentHandler
330
331 attachmentHandlerOne = AttachmentHandler(
332     startingDynamic='mp',
333     endingDynamic='ppp',
334     hairpin='>',
335     articulationList=['flageolet', 'flageolet', 'flageolet', 'flageolet', '',
336     'stopped', 'stopped', '', '', 'flageolet', 'halfopen', 'halfopen', 'halfopen',
337     ', 'flageolet', '', '', 'halfopen', 'flageolet', 'halfopen', 'halfopen',
338     'stopped', 'stopped', 'stopped', 'stopped', 'stopped', 'stopped', 'stopped',
339     'stopped', 'stopped', 'stopped', 'stopped', '', ],
340
341 attachmentHandlerTwo = AttachmentHandler(
342     startingDynamic='p',
343     hairpin='--',
344     articulationList=['stopped'],
345
346 #####sopranino#####
347 sopraninoMusicmakerOne = MusicMaker(
348     rmaker=rmakerOne,
349     pitches=sopraninoRandomWalkNotes,
350     continuous=True,
351     attachmentHandler=attachmentHandlerOne,
352
353 sopraninoMusicmakerTwo = MusicMaker(
354     rmaker=rmakerOne,
355     pitches=sopraninoRandomWalkNotes,
356     continuous=True,
357     attachmentHandler=attachmentHandlerTwo,
358
359 #####soprano_one#####
360 sopranoOneMusicmakerOne = MusicMaker(

```

```
359     rmaker=rmaker_one,
360     pitches=soprano_1_random_walk_notes,
361     continuous=True,
362     attachment_handler=attachment_handler_one,
363 )
364 soprano_one_musicmaker_two = MusicMaker(
365     rmaker=rmaker_one,
366     pitches=soprano_1_random_walk_notes,
367     continuous=True,
368     attachment_handler=attachment_handler_two,
369 )
370 #####soprano_two#####
371 soprano_two_musicmaker_one = MusicMaker(
372     rmaker=rmaker_one,
373     pitches=soprano_2_random_walk_notes,
374     continuous=True,
375     attachment_handler=attachment_handler_one,
376 )
377 soprano_two_musicmaker_two = MusicMaker(
378     rmaker=rmaker_one,
379     pitches=soprano_2_random_walk_notes,
380     continuous=True,
381     attachment_handler=attachment_handler_two,
382 )
383 #####soprano_three#####
384 soprano_three_musicmaker_one = MusicMaker(
385     rmaker=rmaker_one,
386     pitches=soprano_3_random_walk_notes,
387     continuous=True,
388     attachment_handler=attachment_handler_one,
389 )
390 soprano_three_musicmaker_two = MusicMaker(
391     rmaker=rmaker_one,
392     pitches=soprano_3_random_walk_notes,
393     continuous=True,
394     attachment_handler=attachment_handler_two,
395 )
396 #####alto_one#####
397 alto_one_musicmaker_one = MusicMaker(
398     rmaker=rmaker_one,
399     pitches=alto_1_random_walk_notes,
400     continuous=True,
401     attachment_handler=attachment_handler_one,
402 )
403 alto_one_musicmaker_two = MusicMaker(
404     rmaker=rmaker_one,
405     pitches=alto_1_random_walk_notes,
406     continuous=True,
407     attachment_handler=attachment_handler_two,
408 )
409 #####alto_two#####
410 alto_two_musicmaker_one = MusicMaker(
411     rmaker=rmaker_one,
412     pitches=alto_2_random_walk_notes,
413     continuous=True,
414     attachment_handler=attachment_handler_one,
415 )
416 alto_two_musicmaker_two = MusicMaker(
417     rmaker=rmaker_one,
```

```
418     pitches=alto_2_random_walk_notes,
419     continuous=True,
420     attachment_handler=attachment_handler_two,
421 )
422 #####alto_three#####
423 alto_three_musicmaker_one = MusicMaker(
424     rmaker=rmaker_one,
425     pitches=alto_3_random_walk_notes,
426     continuous=True,
427     attachment_handler=attachment_handler_one,
428 )
429 alto_three_musicmaker_two = MusicMaker(
430     rmaker=rmaker_one,
431     pitches=alto_3_random_walk_notes,
432     continuous=True,
433     attachment_handler=attachment_handler_two,
434 )
435 #####alto_four#####
436 alto_four_musicmaker_one = MusicMaker(
437     rmaker=rmaker_one,
438     pitches=alto_4_random_walk_notes,
439     continuous=True,
440     attachment_handler=attachment_handler_one,
441 )
442 alto_four_musicmaker_two = MusicMaker(
443     rmaker=rmaker_one,
444     pitches=alto_4_random_walk_notes,
445     continuous=True,
446     attachment_handler=attachment_handler_two,
447 )
448 #####alto_five#####
449 alto_five_musicmaker_one = MusicMaker(
450     rmaker=rmaker_one,
451     pitches=alto_5_random_walk_notes,
452     continuous=True,
453     attachment_handler=attachment_handler_one,
454 )
455 alto_five_musicmaker_two = MusicMaker(
456     rmaker=rmaker_one,
457     pitches=alto_5_random_walk_notes,
458     continuous=True,
459     attachment_handler=attachment_handler_two,
460 )
461 #####alto_six#####
462 alto_six_musicmaker_one = MusicMaker(
463     rmaker=rmaker_one,
464     pitches=alto_6_random_walk_notes,
465     continuous=True,
466     attachment_handler=attachment_handler_one,
467 )
468 alto_six_musicmaker_two = MusicMaker(
469     rmaker=rmaker_one,
470     pitches=alto_6_random_walk_notes,
471     continuous=True,
472     attachment_handler=attachment_handler_two,
473 )
474 #####tenor_one#####
475 tenor_one_musicmaker_one = MusicMaker(
476     rmaker=rmaker_one,
```

```
477     pitches=tenor_1_random_walk_notes ,
478     continuous=True ,
479     attachment_handler=attachment_handler_one ,
480 )
481 tenor_one_musicmaker_two = MusicMaker(
482     rmaker=rmaker_one ,
483     pitches=tenor_1_random_walk_notes ,
484     continuous=True ,
485     attachment_handler=attachment_handler_two ,
486 )
487 #####tenor_two#####
488 tenor_two_musicmaker_one = MusicMaker(
489     rmaker=rmaker_one ,
490     pitches=tenor_2_random_walk_notes ,
491     continuous=True ,
492     attachment_handler=attachment_handler_one ,
493 )
494 tenor_two_musicmaker_two = MusicMaker(
495     rmaker=rmaker_one ,
496     pitches=tenor_2_random_walk_notes ,
497     continuous=True ,
498     attachment_handler=attachment_handler_two ,
499 )
500 #####tenor_three#####
501 tenor_three_musicmaker_one = MusicMaker(
502     rmaker=rmaker_one ,
503     pitches=tenor_3_random_walk_notes ,
504     continuous=True ,
505     attachment_handler=attachment_handler_one ,
506 )
507 tenor_three_musicmaker_two = MusicMaker(
508     rmaker=rmaker_one ,
509     pitches=tenor_3_random_walk_notes ,
510     continuous=True ,
511     attachment_handler=attachment_handler_two ,
512 )
513 #####tenor_four#####
514 tenor_four_musicmaker_one = MusicMaker(
515     rmaker=rmaker_one ,
516     pitches=tenor_4_random_walk_notes ,
517     continuous=True ,
518     attachment_handler=attachment_handler_one ,
519 )
520 tenor_four_musicmaker_two = MusicMaker(
521     rmaker=rmaker_one ,
522     pitches=tenor_4_random_walk_notes ,
523     continuous=True ,
524     attachment_handler=attachment_handler_two ,
525 )
526 #####tenor_five#####
527 tenor_five_musicmaker_one = MusicMaker(
528     rmaker=rmaker_one ,
529     pitches=tenor_5_random_walk_notes ,
530     continuous=True ,
531     attachment_handler=attachment_handler_one ,
532 )
533 tenor_five_musicmaker_two = MusicMaker(
534     rmaker=rmaker_one ,
535     pitches=tenor_5_random_walk_notes ,
```

```
536     continuous=True,
537     attachment_handler=attachment_handler_two,
538 )
539 #####baritone_one#####
540 baritone_one_musicmaker_one = MusicMaker(
541     rmaker=rmaker_one,
542     pitches=baritone_1_random_walk_notes,
543     continuous=True,
544     attachment_handler=attachment_handler_one,
545 )
546 baritone_one_musicmaker_two = MusicMaker(
547     rmaker=rmaker_one,
548     pitches=baritone_1_random_walk_notes,
549     continuous=True,
550     attachment_handler=attachment_handler_two,
551 )
552 #####baritone_two#####
553 baritone_two_musicmaker_one = MusicMaker(
554     rmaker=rmaker_one,
555     pitches=baritone_2_random_walk_notes,
556     continuous=True,
557     attachment_handler=attachment_handler_one,
558 )
559 baritone_two_musicmaker_two = MusicMaker(
560     rmaker=rmaker_one,
561     pitches=baritone_2_random_walk_notes,
562     continuous=True,
563     attachment_handler=attachment_handler_two,
564 )
565 #####baritone_three#####
566 baritone_three_musicmaker_one = MusicMaker(
567     rmaker=rmaker_one,
568     pitches=baritone_3_random_walk_notes,
569     continuous=True,
570     attachment_handler=attachment_handler_one,
571 )
572 baritone_three_musicmaker_two = MusicMaker(
573     rmaker=rmaker_one,
574     pitches=baritone_3_random_walk_notes,
575     continuous=True,
576     attachment_handler=attachment_handler_two,
577 )
578 #####bass_one#####
579 bass_one_musicmaker_one = MusicMaker(
580     rmaker=rmaker_one,
581     pitches=bass_1_random_walk_notes,
582     continuous=True,
583     attachment_handler=attachment_handler_one,
584 )
585 bass_one_musicmaker_two = MusicMaker(
586     rmaker=rmaker_one,
587     pitches=bass_1_random_walk_notes,
588     continuous=True,
589     attachment_handler=attachment_handler_two,
590 )
591 #####bass_two#####
592 bass_two_musicmaker_one = MusicMaker(
593     rmaker=rmaker_one,
594     pitches=bass_2_random_walk_notes,
```

```

595     continuous=True,
596     attachment_handler=attachment_handler_one,
597 )
598 bass_two_musicmaker_two = MusicMaker(
599     rmaker=rmaker_one,
600     pitches=bass_2_random_walk_notes,
601     continuous=True,
602     attachment_handler=attachment_handler_two,
603 )
604 #####contrabass#####
605 contrabass_musicmaker_one = MusicMaker(
606     rmaker=rmaker_one,
607     pitches=contrabass_random_walk_notes,
608     continuous=True,
609     attachment_handler=attachment_handler_one,
610 )
611 contrabass_musicmaker_two = MusicMaker(
612     rmaker=rmaker_one,
613     pitches=contrabass_random_walk_notes,
614     continuous=True,
615     attachment_handler=attachment_handler_two,
616 )
617
618 silence_maker = abjadext.rmakers.NoteRhythmMaker(
619     division_masks=[
620         abjadext.rmakers.SilenceMask(
621             pattern=abjad.index([0], 1),
622             ),
623         ],
624     )
625
626 class MusicSpecifier:
627
628     def __init__(self, music_maker, voice_name):
629         self.music_maker = music_maker
630         self.voice_name = voice_name
631
632     print('Collecting timespans and rmakers ...')
633
634 voice_1_timespan_list = abjad.TimespanList([
635     abjad.AnnotatedTimespan(
636         start_offset=start_offset,
637         stop_offset=stop_offset,
638         annotation=MusicSpecifier(
639             music_maker=music_maker,
640             voice_name='Voice 1',
641             ),
642         )
643     for start_offset, stop_offset, music_maker in [
644         [(0, 8), (2, 8), soprano_musicmaker_one],
645         [(2, 8), (4, 8), soprano_musicmaker_one],
646         [(4, 8), (6, 8), soprano_musicmaker_two],
647         [(6, 8), (8, 8), soprano_musicmaker_one],
648
649         [(22, 8), (24, 8), soprano_musicmaker_two],
650         [(24, 8), (26, 8), soprano_musicmaker_two],
651         [(26, 8), (28, 8), soprano_musicmaker_one],
652
653         [(34, 8), (36, 8), soprano_musicmaker_one],

```

```

654     [(36, 8), (38, 8), soprano_musicmaker_two],
655     [(38, 8), (40, 8), soprano_musicmaker_one],
656     [(40, 8), (42, 8), soprano_musicmaker_two],
657
658     [(50, 8), (52, 8), soprano_musicmaker_one],
659     [(52, 8), (54, 8), soprano_musicmaker_two],
660     [(54, 8), (56, 8), soprano_musicmaker_one],
661
662     [(58, 8), (60, 8), soprano_musicmaker_two],
663
664     [(62, 8), (64, 8), soprano_musicmaker_one],
665     [(64, 8), (66, 8), soprano_musicmaker_one],
666
667     [(68, 8), (70, 8), soprano_musicmaker_one],
668     [(70, 8), (72, 8), soprano_musicmaker_two],
669     [(72, 8), (74, 8), soprano_musicmaker_two],
670
671     [(76, 8), (78, 8), soprano_musicmaker_one],
672     [(78, 8), (80, 8), soprano_musicmaker_two],
673     [(80, 8), (82, 8), soprano_musicmaker_one],
674     [(82, 8), (84, 8), soprano_musicmaker_two],
675
676     [(92, 8), (94, 8), soprano_musicmaker_one],
677     [(94, 8), (96, 8), soprano_musicmaker_one],
678     [(96, 8), (98, 8), soprano_musicmaker_two],
679     [(98, 8), (100, 8), soprano_musicmaker_one],
680     [(100, 8), (102, 8), soprano_musicmaker_one],
681     [(102, 8), (104, 8), soprano_musicmaker_two],
682     [(104, 8), (106, 8), soprano_musicmaker_two],
683     [(106, 8), (108, 8), soprano_musicmaker_one],
684     [(108, 8), (110, 8), soprano_musicmaker_two],
685 ]
686 ])
687
688 voice_2_timespan_list = abjad.TimespanList([
689     abjad.AnnotatedTimespan(
690         start_offset=start_offset,
691         stop_offset=stop_offset,
692         annotation=MusicSpecifier(
693             music_maker=music_maker,
694             voice_name='Voice 2',
695         ),
696     )
697     for start_offset, stop_offset, music_maker in [
698         [(0, 8), (2, 8), soprano_one_musicmaker_two],
699         [(3, 8), (4, 8), soprano_one_musicmaker_two],
700         [(4, 8), (5, 8), soprano_one_musicmaker_two],
701         [(7, 8), (8, 8), soprano_one_musicmaker_one],
702         [(9, 8), (10, 8), soprano_one_musicmaker_two],
703         [(10, 8), (11, 8), soprano_one_musicmaker_two],
704
705         [(43, 8), (44, 8), soprano_one_musicmaker_one],
706         [(45, 8), (46, 8), soprano_one_musicmaker_one],
707         [(47, 8), (48, 8), soprano_one_musicmaker_two],
708
709         [(63, 8), (64, 8), soprano_one_musicmaker_two],
710         [(65, 8), (66, 8), soprano_one_musicmaker_two],
711         [(67, 8), (68, 8), soprano_one_musicmaker_one],
712         [(69, 8), (70, 8), soprano_one_musicmaker_two],

```

```

713     [(71, 8), (72, 8), soprano_one_musicmaker_one],
714     [(73, 8), (74, 8), soprano_one_musicmaker_two],
715
716     [(76, 8), (77, 8), soprano_one_musicmaker_two],
717     [(78, 8), (79, 8), soprano_one_musicmaker_one],
718     [(80, 8), (81, 8), soprano_one_musicmaker_one],
719
720     [(88, 8), (90, 8), soprano_one_musicmaker_one],
721     [(90, 8), (92, 8), soprano_one_musicmaker_one],
722     [(92, 8), (94, 8), soprano_one_musicmaker_two],
723     [(94, 8), (96, 8), soprano_one_musicmaker_one],
724     [(96, 8), (98, 8), soprano_one_musicmaker_two],
725
726     [(101, 8), (102, 8), soprano_one_musicmaker_two],
727     [(103, 8), (104, 8), soprano_one_musicmaker_two],
728     [(105, 8), (106, 8), soprano_one_musicmaker_one],
729
730     [(110, 8), (111, 8), soprano_one_musicmaker_one],
731     [(112, 8), (113, 8), soprano_one_musicmaker_one],
732     [(114, 8), (115, 8), soprano_one_musicmaker_two],
733     [(116, 8), (117, 8), soprano_one_musicmaker_one],
734     [(118, 8), (119, 8), soprano_one_musicmaker_one],
735 ]
736 ])
737
738 voice_3_timespan_list = abjad.TimespanList([
739     abjad.AnnotatedTimespan(
740         start_offset=start_offset,
741         stop_offset=stop_offset,
742         annotation=MusicSpecifier(
743             music_maker=music_maker,
744             voice_name='Voice 3',
745         ),
746     ),
747     for start_offset, stop_offset, music_maker in [
748
749         [(6, 8), (8, 8), soprano_two_musicmaker_one],
750         [(8, 8), (10, 8), soprano_two_musicmaker_two],
751
752         [(22, 8), (24, 8), soprano_two_musicmaker_one],
753         [(24, 8), (26, 8), soprano_two_musicmaker_two],
754         [(26, 8), (28, 8), soprano_two_musicmaker_two],
755         [(28, 8), (30, 8), soprano_two_musicmaker_one],
756         [(30, 8), (32, 8), soprano_two_musicmaker_two],
757
758         [(38, 8), (40, 8), soprano_two_musicmaker_two],
759         [(40, 8), (42, 8), soprano_two_musicmaker_two],
760         [(42, 8), (44, 8), soprano_two_musicmaker_one],
761
762         [(48, 8), (50, 8), soprano_two_musicmaker_one],
763
764         [(60, 8), (62, 8), soprano_two_musicmaker_two],
765         [(62, 8), (64, 8), soprano_two_musicmaker_one],
766         [(64, 8), (66, 8), soprano_two_musicmaker_two],
767         [(66, 8), (68, 8), soprano_two_musicmaker_two],
768
769         [(78, 8), (80, 8), soprano_two_musicmaker_two],
770         [(80, 8), (82, 8), soprano_two_musicmaker_one],
771         [(82, 8), (84, 8), soprano_two_musicmaker_two],

```

```

772
773     [(90, 8), (92, 8), soprano_two_musicmaker_one],
774     [(92, 8), (94, 8), soprano_two_musicmaker_two],
775     [(94, 8), (96, 8), soprano_two_musicmaker_one],
776     [(96, 8), (98, 8), soprano_two_musicmaker_one],
777     [(98, 8), (100, 8), soprano_two_musicmaker_two],
778     [(100, 8), (102, 8), soprano_two_musicmaker_two],
779
780     [(114, 8), (116, 8), soprano_two_musicmaker_one],
781     [(116, 8), (118, 8), soprano_two_musicmaker_two],
782     [(118, 8), (120, 8), soprano_two_musicmaker_one],
783 ]
784 ])
785
786 voice_4_timespan_list = abjad.TimespanList([
787     abjad.AnnotatedTimespan(
788         start_offset=start_offset,
789         stop_offset=stop_offset,
790         annotation=MusicSpecifier(
791             music_maker=music_maker,
792             voice_name='Voice 4',
793         ),
794     )
795     for start_offset, stop_offset, music_maker in [
796         [(0, 8), (2, 8), soprano_three_musicmaker_one],
797         [(2, 8), (4, 8), soprano_three_musicmaker_one],
798         [(4, 8), (6, 8), soprano_three_musicmaker_one],
799
800         [(8, 8), (10, 8), soprano_three_musicmaker_one],
801         [(10, 8), (12, 8), soprano_three_musicmaker_one],
802
803         [(38, 8), (40, 8), soprano_three_musicmaker_two],
804         [(40, 8), (42, 8), soprano_three_musicmaker_two],
805
806         [(44, 8), (46, 8), soprano_three_musicmaker_two],
807         [(46, 8), (48, 8), soprano_three_musicmaker_one],
808         [(48, 8), (50, 8), soprano_three_musicmaker_one],
809         [(50, 8), (52, 8), soprano_three_musicmaker_one],
810
811         [(54, 8), (56, 8), soprano_three_musicmaker_two],
812         [(56, 8), (58, 8), soprano_three_musicmaker_one],
813
814         [(76, 8), (78, 8), soprano_three_musicmaker_two],
815         [(78, 8), (80, 8), soprano_three_musicmaker_one],
816         [(80, 8), (82, 8), soprano_three_musicmaker_one],
817         [(82, 8), (84, 8), soprano_three_musicmaker_two],
818         [(84, 8), (86, 8), soprano_three_musicmaker_one],
819         [(86, 8), (88, 8), soprano_three_musicmaker_one],
820
821         [(90, 8), (92, 8), soprano_three_musicmaker_two],
822
823         [(94, 8), (96, 8), soprano_three_musicmaker_one],
824         [(96, 8), (98, 8), soprano_three_musicmaker_one],
825
826         [(114, 8), (116, 8), soprano_three_musicmaker_one],
827         [(116, 8), (118, 8), soprano_three_musicmaker_two],
828         [(118, 8), (120, 8), soprano_three_musicmaker_two],
829     ]
830 ])

```

```

831
832 voice_5_timespan_list = abjad.TimespanList([
833     abjad.AnnotatedTimespan(
834         start_offset=start_offset,
835         stop_offset=stop_offset,
836         annotation=MusicSpecifier(
837             music_maker=music_maker,
838             voice_name='Voice 5',
839         ),
840     )
841     for start_offset, stop_offset, music_maker in [
842         [(14, 8), (16, 8), alto_one_musicmaker_two],
843         [(16, 8), (18, 8), alto_one_musicmaker_two],
844
845         [(22, 8), (24, 8), alto_one_musicmaker_two],
846         [(24, 8), (26, 8), alto_one_musicmaker_two],
847         [(26, 8), (28, 8), alto_one_musicmaker_two],
848         [(28, 8), (30, 8), alto_one_musicmaker_one],
849         [(30, 8), (32, 8), alto_one_musicmaker_two],
850
851         [(38, 8), (40, 8), alto_one_musicmaker_one],
852         [(40, 8), (42, 8), alto_one_musicmaker_two],
853         [(42, 8), (44, 8), alto_one_musicmaker_two],
854         [(44, 8), (46, 8), alto_one_musicmaker_one],
855         [(46, 8), (48, 8), alto_one_musicmaker_one],
856         [(48, 8), (50, 8), alto_one_musicmaker_two],
857         [(50, 8), (52, 8), alto_one_musicmaker_two],
858
859         [(54, 8), (56, 8), alto_one_musicmaker_one],
860         [(56, 8), (58, 8), alto_one_musicmaker_one],
861
862         [(64, 8), (66, 8), alto_one_musicmaker_one],
863         [(66, 8), (68, 8), alto_one_musicmaker_two],
864         [(68, 8), (70, 8), alto_one_musicmaker_one],
865         [(70, 8), (72, 8), alto_one_musicmaker_one],
866         [(72, 8), (74, 8), alto_one_musicmaker_two],
867
868         [(76, 8), (78, 8), alto_one_musicmaker_two],
869         [(78, 8), (80, 8), alto_one_musicmaker_one],
870
871         [(82, 8), (84, 8), alto_one_musicmaker_one],
872         [(84, 8), (86, 8), alto_one_musicmaker_two],
873         [(86, 8), (88, 8), alto_one_musicmaker_one],
874
875         [(92, 8), (94, 8), alto_one_musicmaker_two],
876         [(94, 8), (96, 8), alto_one_musicmaker_one],
877         [(96, 8), (98, 8), alto_one_musicmaker_two],
878         [(98, 8), (100, 8), alto_one_musicmaker_one],
879         [(100, 8), (102, 8), alto_one_musicmaker_two],
880
881         [(104, 8), (106, 8), alto_one_musicmaker_two],
882         [(106, 8), (108, 8), alto_one_musicmaker_one],
883         [(108, 8), (110, 8), alto_one_musicmaker_two],
884         [(110, 8), (112, 8), alto_one_musicmaker_two],
885
886         [(114, 8), (116, 8), alto_one_musicmaker_one],
887         [(116, 8), (118, 8), alto_one_musicmaker_two],
888         [(118, 8), (120, 8), alto_one_musicmaker_one],
889     ]

```

```
890 ])
891
892 voice_6_timespan_list = abjad.TimespanList([
893     abjad.AnnotatedTimespan(
894         start_offset=start_offset,
895         stop_offset=stop_offset,
896         annotation=MusicSpecifier(
897             music_maker=music_maker,
898             voice_name='Voice 6',
899         ),
900     )
901     for start_offset, stop_offset, music_maker in [
902
903         [(10, 8), (12, 8), alto_two_musicmaker_one],
904         [(12, 8), (14, 8), alto_two_musicmaker_two],
905         [(14, 8), (16, 8), alto_two_musicmaker_two],
906         [(16, 8), (18, 8), alto_two_musicmaker_one],
907         [(18, 8), (20, 8), alto_two_musicmaker_one],
908         [(20, 8), (22, 8), alto_two_musicmaker_one],
909
910         [(30, 8), (32, 8), alto_two_musicmaker_one],
911         [(32, 8), (34, 8), alto_two_musicmaker_one],
912         [(34, 8), (36, 8), alto_two_musicmaker_one],
913         [(36, 8), (38, 8), alto_two_musicmaker_two],
914         [(38, 8), (40, 8), alto_two_musicmaker_two],
915
916         [(42, 8), (44, 8), alto_two_musicmaker_one],
917         [(44, 8), (46, 8), alto_two_musicmaker_one],
918         [(46, 8), (48, 8), alto_two_musicmaker_two],
919
920         [(50, 8), (52, 8), alto_two_musicmaker_two],
921         [(52, 8), (54, 8), alto_two_musicmaker_one],
922         [(54, 8), (56, 8), alto_two_musicmaker_one],
923
924         [(58, 8), (60, 8), alto_two_musicmaker_one],
925
926         [(62, 8), (64, 8), alto_two_musicmaker_two],
927
928         [(66, 8), (68, 8), alto_two_musicmaker_one],
929         [(68, 8), (70, 8), alto_two_musicmaker_two],
930
931         [(74, 8), (76, 8), alto_two_musicmaker_two],
932         [(76, 8), (78, 8), alto_two_musicmaker_one],
933         [(78, 8), (80, 8), alto_two_musicmaker_one],
934         [(80, 8), (82, 8), alto_two_musicmaker_two],
935         [(82, 8), (84, 8), alto_two_musicmaker_two],
936         [(84, 8), (86, 8), alto_two_musicmaker_one],
937
938         [(88, 8), (90, 8), alto_two_musicmaker_one],
939         [(90, 8), (92, 8), alto_two_musicmaker_two],
940
941         [(94, 8), (96, 8), alto_two_musicmaker_one],
942         [(96, 8), (98, 8), alto_two_musicmaker_one],
943         [(98, 8), (100, 8), alto_two_musicmaker_two],
944         [(100, 8), (102, 8), alto_two_musicmaker_two],
945
946         [(114, 8), (116, 8), alto_two_musicmaker_two],
947         [(116, 8), (118, 8), alto_two_musicmaker_one],
948         [(118, 8), (120, 8), alto_two_musicmaker_two],
```

```
949     ]
950 ])
951
952 voice_7_timespan_list = abjad.TimespanList([
953     abjad.AnnotatedTimespan(
954         start_offset=start_offset,
955         stop_offset=stop_offset,
956         annotation=MusicSpecifier(
957             music_maker=music_maker,
958             voice_name='Voice 7',
959         ),
960     )
961     for start_offset, stop_offset, music_maker in [
962         [(0, 8), (2, 8), alto_three_musicmaker_two],
963         [(2, 8), (4, 8), alto_three_musicmaker_one],
964         [(4, 8), (6, 8), alto_three_musicmaker_one],
965         [(6, 8), (8, 8), alto_three_musicmaker_two],
966
967         [(14, 8), (16, 8), alto_three_musicmaker_one],
968         [(16, 8), (18, 8), alto_three_musicmaker_one],
969         [(18, 8), (20, 8), alto_three_musicmaker_one],
970
971         [(22, 8), (24, 8), alto_three_musicmaker_two],
972         [(24, 8), (26, 8), alto_three_musicmaker_one],
973         [(26, 8), (28, 8), alto_three_musicmaker_two],
974
975         [(34, 8), (36, 8), alto_three_musicmaker_one],
976         [(36, 8), (38, 8), alto_three_musicmaker_two],
977
978         [(42, 8), (44, 8), alto_three_musicmaker_one],
979         [(44, 8), (46, 8), alto_three_musicmaker_one],
980         [(46, 8), (48, 8), alto_three_musicmaker_one],
981
982         [(52, 8), (54, 8), alto_three_musicmaker_one],
983         [(54, 8), (56, 8), alto_three_musicmaker_one],
984
985         [(60, 8), (62, 8), alto_three_musicmaker_two],
986
987         [(64, 8), (66, 8), alto_three_musicmaker_one],
988         [(66, 8), (68, 8), alto_three_musicmaker_two],
989         [(68, 8), (70, 8), alto_three_musicmaker_two],
990         [(70, 8), (72, 8), alto_three_musicmaker_two],
991
992         [(80, 8), (82, 8), alto_three_musicmaker_two],
993
994         [(86, 8), (88, 8), alto_three_musicmaker_two],
995         [(88, 8), (90, 8), alto_three_musicmaker_two],
996
997         [(92, 8), (94, 8), alto_three_musicmaker_one],
998         [(94, 8), (96, 8), alto_three_musicmaker_one],
999         [(96, 8), (98, 8), alto_three_musicmaker_two],
1000        [(98, 8), (100, 8), alto_three_musicmaker_two],
1001        [(100, 8), (102, 8), alto_three_musicmaker_two],
1002
1003        [(104, 8), (106, 8), alto_three_musicmaker_one],
1004        [(106, 8), (108, 8), alto_three_musicmaker_two],
1005
1006        [(110, 8), (112, 8), alto_three_musicmaker_one],
1007    ])
```

```

1008     ],
1009   ]
1010 ])
1011
1012 voice_8_timespan_list = abjad.TimespanList([
1013     abjad.AnnotatedTimespan(
1014         start_offset=start_offset,
1015         stop_offset=stop_offset,
1016         annotation=MusicSpecifier(
1017             music_maker=music_maker,
1018             voice_name='Voice 8',
1019         ),
1020     )
1021     for start_offset, stop_offset, music_maker in [
1022         [(1, 8), (2, 8), alto_three_musicmaker_one],
1023
1024         [(5, 8), (6, 8), alto_four_musicmaker_two],
1025
1026         [(9, 8), (10, 8), alto_four_musicmaker_two],
1027         [(11, 8), (12, 8), alto_four_musicmaker_two],
1028
1029         [(15, 8), (16, 8), alto_four_musicmaker_one],
1030         [(17, 8), (18, 8), alto_four_musicmaker_one],
1031         [(19, 8), (20, 8), alto_four_musicmaker_two],
1032         [(21, 8), (22, 8), alto_four_musicmaker_one],
1033
1034         [(24, 8), (25, 8), alto_four_musicmaker_two],
1035         [(26, 8), (27, 8), alto_four_musicmaker_two],
1036         [(28, 8), (29, 8), alto_four_musicmaker_one],
1037
1038         [(31, 8), (34, 8), alto_four_musicmaker_two],
1039         [(35, 8), (36, 8), alto_four_musicmaker_two],
1040         [(37, 8), (38, 8), alto_four_musicmaker_one],
1041
1042         [(41, 8), (42, 8), alto_four_musicmaker_one],
1043         [(43, 8), (44, 8), alto_four_musicmaker_two],
1044         [(45, 8), (46, 8), alto_four_musicmaker_one],
1045         [(47, 8), (48, 8), alto_four_musicmaker_two],
1046         [(49, 8), (50, 8), alto_four_musicmaker_two],
1047
1048         [(54, 8), (56, 8), alto_four_musicmaker_one],
1049
1050         [(58, 8), (60, 8), alto_four_musicmaker_two],
1051         [(60, 8), (62, 8), alto_four_musicmaker_two],
1052         [(62, 8), (64, 8), alto_four_musicmaker_one],
1053
1054         [(66, 8), (68, 8), alto_four_musicmaker_two],
1055         [(68, 8), (70, 8), alto_four_musicmaker_two],
1056         [(70, 8), (72, 8), alto_four_musicmaker_one],
1057         [(72, 8), (74, 8), alto_four_musicmaker_two],
1058
1059         [(76, 8), (78, 8), alto_four_musicmaker_one],
1060         [(78, 8), (80, 8), alto_four_musicmaker_two],
1061         [(80, 8), (82, 8), alto_four_musicmaker_two],
1062
1063         [(85, 8), (86, 8), alto_four_musicmaker_one],
1064         [(87, 8), (88, 8), alto_four_musicmaker_one],
1065         [(89, 8), (90, 8), alto_four_musicmaker_two],
1066         [(91, 8), (92, 8), alto_four_musicmaker_two],

```

```

1067     [(96, 8), (98, 8), alto_four_musicmaker_two],
1068     [(98, 8), (100, 8), alto_four_musicmaker_two],
1069     [(100, 8), (102, 8), alto_four_musicmaker_one],
1070     [(102, 8), (104, 8), alto_four_musicmaker_two],
1071     [(104, 8), (106, 8), alto_four_musicmaker_one],
1072     [(106, 8), (108, 8), alto_four_musicmaker_one],
1073
1074     [(110, 8), (112, 8), alto_four_musicmaker_two],
1075     [(112, 8), (114, 8), alto_four_musicmaker_one],
1076
1077     [(116, 8), (118, 8), alto_four_musicmaker_two],
1078     [(118, 8), (120, 8), alto_four_musicmaker_two],
1079
1080 ]
1081 ])
1082
1083 voice_9_timespan_list = abjad.TimespanList([
1084     abjad.AnnotatedTimespan(
1085         start_offset=start_offset,
1086         stop_offset=stop_offset,
1087         annotation=MusicSpecifier(
1088             music_maker=music_maker,
1089             voice_name='Voice 9',
1090         ),
1091     ),
1092     for start_offset, stop_offset, music_maker in [
1093         [(0, 8), (2, 8), alto_five_musicmaker_one],
1094         [(2, 8), (4, 8), alto_five_musicmaker_one],
1095
1096         [(8, 8), (10, 8), alto_five_musicmaker_one],
1097         [(10, 8), (12, 8), alto_five_musicmaker_two],
1098         [(12, 8), (14, 8), alto_five_musicmaker_two],
1099
1100         [(28, 8), (30, 8), alto_five_musicmaker_two],
1101         [(30, 8), (32, 8), alto_five_musicmaker_two],
1102         [(32, 8), (34, 8), alto_five_musicmaker_one],
1103         [(34, 8), (36, 8), alto_five_musicmaker_one],
1104
1105         [(38, 8), (40, 8), alto_five_musicmaker_two],
1106
1107         [(42, 8), (44, 8), alto_five_musicmaker_one],
1108         [(44, 8), (46, 8), alto_five_musicmaker_two],
1109
1110         [(62, 8), (64, 8), alto_five_musicmaker_two],
1111         [(64, 8), (66, 8), alto_five_musicmaker_one],
1112         [(66, 8), (68, 8), alto_five_musicmaker_two],
1113
1114         [(70, 8), (72, 8), alto_five_musicmaker_one],
1115         [(72, 8), (74, 8), alto_five_musicmaker_two],
1116         [(74, 8), (76, 8), alto_five_musicmaker_two],
1117
1118         [(96, 8), (98, 8), alto_five_musicmaker_two],
1119         [(98, 8), (100, 8), alto_five_musicmaker_two],
1120         [(100, 8), (102, 8), alto_five_musicmaker_two],
1121         [(102, 8), (104, 8), alto_five_musicmaker_one],
1122         [(104, 8), (106, 8), alto_five_musicmaker_one],
1123
1124         [(108, 8), (110, 8), alto_five_musicmaker_two],
1125

```

```
1126     [(110, 8), (112, 8), alto_five_musicmaker_two],
1127     [(112, 8), (114, 8), alto_five_musicmaker_two],
1128 ]
1129 ])
1130
1131 voice_10_timestspan_list = abjad.TimespanList([
1132     abjad.AnnotatedTimespan(
1133         start_offset=start_offset,
1134         stop_offset=stop_offset,
1135         annotation=MusicSpecifier(
1136             music_maker=music_maker,
1137             voice_name='Voice 10',
1138         ),
1139     )
1140     for start_offset, stop_offset, music_maker in [
1141         [(0, 8), (2, 8), alto_six_musicmaker_two],
1142         [(2, 8), (4, 8), alto_six_musicmaker_one],
1143
1144         [(6, 8), (8, 8), alto_six_musicmaker_two],
1145         [(8, 8), (10, 8), alto_six_musicmaker_one],
1146         [(10, 8), (12, 8), alto_six_musicmaker_one],
1147         [(12, 8), (14, 8), alto_six_musicmaker_one],
1148
1149         [(16, 8), (18, 8), alto_six_musicmaker_two],
1150         [(18, 8), (20, 8), alto_six_musicmaker_one],
1151
1152         [(22, 8), (24, 8), alto_six_musicmaker_one],
1153         [(24, 8), (26, 8), alto_six_musicmaker_one],
1154
1155         [(28, 8), (30, 8), alto_six_musicmaker_two],
1156         [(30, 8), (32, 8), alto_six_musicmaker_one],
1157         [(32, 8), (34, 8), alto_six_musicmaker_one],
1158
1159         [(36, 8), (38, 8), alto_six_musicmaker_two],
1160         [(38, 8), (40, 8), alto_six_musicmaker_one],
1161
1162         [(42, 8), (44, 8), alto_six_musicmaker_one],
1163         [(44, 8), (46, 8), alto_six_musicmaker_two],
1164         [(46, 8), (48, 8), alto_six_musicmaker_one],
1165         [(48, 8), (50, 8), alto_six_musicmaker_one],
1166         [(50, 8), (52, 8), alto_six_musicmaker_one],
1167
1168         [(54, 8), (56, 8), alto_six_musicmaker_one],
1169         [(56, 8), (58, 8), alto_six_musicmaker_one],
1170
1171
1172         [(62, 8), (64, 8), alto_six_musicmaker_two],
1173         [(64, 8), (66, 8), alto_six_musicmaker_two],
1174         [(66, 8), (68, 8), alto_six_musicmaker_one],
1175
1176         [(70, 8), (72, 8), alto_six_musicmaker_one],
1177         [(72, 8), (74, 8), alto_six_musicmaker_two],
1178         [(74, 8), (76, 8), alto_six_musicmaker_two],
1179
1180         [(78, 8), (80, 8), alto_six_musicmaker_one],
1181
1182         [(82, 8), (84, 8), alto_six_musicmaker_one],
1183         [(84, 8), (86, 8), alto_six_musicmaker_one],
1184         [(86, 8), (88, 8), alto_six_musicmaker_two],
```

```

1185     [(90, 8), (92, 8), alto_six_musicmaker_two],
1186     [(92, 8), (94, 8), alto_six_musicmaker_two],
1187
1188     [(96, 8), (98, 8), alto_six_musicmaker_one],
1189     [(98, 8), (100, 8), alto_six_musicmaker_one],
1190     [(100, 8), (102, 8), alto_six_musicmaker_two],
1191     [(102, 8), (104, 8), alto_six_musicmaker_two],
1192     [(104, 8), (106, 8), alto_six_musicmaker_one],
1193
1194     [(108, 8), (110, 8), alto_six_musicmaker_one],
1195     [(110, 8), (112, 8), alto_six_musicmaker_two],
1196     [(112, 8), (114, 8), alto_six_musicmaker_one],
1197
1198 ]
1199 ])
1200
1201 voice_11_timespan_list = abjad.TimespanList([
1202     abjad.AnnotatedTimespan(
1203         start_offset=start_offset,
1204         stop_offset=stop_offset,
1205         annotation=MusicSpecifier(
1206             music_maker=music_maker,
1207             voice_name='Voice 11',
1208         ),
1209     )
1210     for start_offset, stop_offset, music_maker in [
1211         [(0, 8), (2, 8), tenor_one_musicmaker_one],
1212         [(2, 8), (4, 8), tenor_one_musicmaker_one],
1213
1214         [(6, 8), (8, 8), tenor_one_musicmaker_one],
1215         [(8, 8), (10, 8), tenor_one_musicmaker_one],
1216         [(10, 8), (12, 8), tenor_one_musicmaker_two],
1217         [(12, 8), (14, 8), tenor_one_musicmaker_one],
1218
1219         [(16, 8), (18, 8), tenor_one_musicmaker_one],
1220         [(18, 8), (20, 8), tenor_one_musicmaker_one],
1221
1222         [(22, 8), (24, 8), tenor_one_musicmaker_two],
1223         [(24, 8), (26, 8), tenor_one_musicmaker_one],
1224         [(26, 8), (28, 8), tenor_one_musicmaker_one],
1225         [(28, 8), (30, 8), tenor_one_musicmaker_one],
1226         [(30, 8), (32, 8), tenor_one_musicmaker_one],
1227         [(32, 8), (34, 8), tenor_one_musicmaker_two],
1228         [(34, 8), (36, 8), tenor_one_musicmaker_two],
1229
1230         [(44, 8), (46, 8), tenor_one_musicmaker_two],
1231         [(46, 8), (48, 8), tenor_one_musicmaker_two],
1232         [(48, 8), (50, 8), tenor_one_musicmaker_one],
1233         [(50, 8), (52, 8), tenor_one_musicmaker_two],
1234         [(52, 8), (54, 8), tenor_one_musicmaker_two],
1235         [(54, 8), (56, 8), tenor_one_musicmaker_one],
1236
1237         [(60, 8), (62, 8), tenor_one_musicmaker_two],
1238         [(62, 8), (64, 8), tenor_one_musicmaker_one],
1239
1240         [(66, 8), (68, 8), tenor_one_musicmaker_two],
1241         [(68, 8), (70, 8), tenor_one_musicmaker_two],
1242         [(70, 8), (72, 8), tenor_one_musicmaker_one],
1243

```

```

1244     [(74, 8), (76, 8), tenor_one_musicmaker_one],
1245     [(76, 8), (78, 8), tenor_one_musicmaker_two],
1246
1247     [(82, 8), (84, 8), tenor_one_musicmaker_one],
1248     [(84, 8), (86, 8), tenor_one_musicmaker_two],
1249
1250     [(88, 8), (90, 8), tenor_one_musicmaker_two],
1251     [(90, 8), (92, 8), tenor_one_musicmaker_one],
1252     [(92, 8), (94, 8), tenor_one_musicmaker_one],
1253     [(94, 8), (96, 8), tenor_one_musicmaker_two],
1254     [(96, 8), (98, 8), tenor_one_musicmaker_two],
1255     [(98, 8), (100, 8), tenor_one_musicmaker_one],
1256
1257     [(102, 8), (104, 8), tenor_one_musicmaker_two],
1258     [(104, 8), (106, 8), tenor_one_musicmaker_two],
1259
1260     [(110, 8), (112, 8), tenor_one_musicmaker_two],
1261     [(112, 8), (114, 8), tenor_one_musicmaker_one],
1262     [(114, 8), (116, 8), tenor_one_musicmaker_one],
1263     [(116, 8), (118, 8), tenor_one_musicmaker_two],
1264 ]
1265 ])
1266
1267 voice_12_timespan_list = abjad.TimespanList([
1268     abjad.AnnotatedTimespan(
1269         start_offset=start_offset,
1270         stop_offset=stop_offset,
1271         annotation=MusicSpecifier(
1272             music_maker=music_maker,
1273             voice_name='Voice 12',
1274         ),
1275     ),
1276     for start_offset, stop_offset, music_maker in [
1277
1278         [(6, 8), (8, 8), tenor_two_musicmaker_one],
1279         [(8, 8), (10, 8), tenor_two_musicmaker_two],
1280         [(10, 8), (12, 8), tenor_two_musicmaker_one],
1281         [(12, 8), (14, 8), tenor_two_musicmaker_two],
1282         [(14, 8), (16, 8), tenor_two_musicmaker_one],
1283
1284         [(18, 8), (20, 8), tenor_two_musicmaker_one],
1285
1286         [(26, 8), (28, 8), tenor_two_musicmaker_one],
1287         [(28, 8), (30, 8), tenor_two_musicmaker_two],
1288         [(30, 8), (32, 8), tenor_two_musicmaker_two],
1289
1290         [(34, 8), (36, 8), tenor_two_musicmaker_one],
1291         [(36, 8), (38, 8), tenor_two_musicmaker_one],
1292         [(38, 8), (40, 8), tenor_two_musicmaker_two],
1293         [(40, 8), (42, 8), tenor_two_musicmaker_two],
1294         [(42, 8), (44, 8), tenor_two_musicmaker_one],
1295         [(44, 8), (46, 8), tenor_two_musicmaker_one],
1296         [(46, 8), (48, 8), tenor_two_musicmaker_two],
1297         [(48, 8), (50, 8), tenor_two_musicmaker_two],
1298         [(50, 8), (52, 8), tenor_two_musicmaker_one],
1299
1300         [(54, 8), (56, 8), tenor_two_musicmaker_two],
1301         [(56, 8), (58, 8), tenor_two_musicmaker_one],
1302

```

```

1303     [(60, 8), (62, 8), tenor_two_musicmaker_two],
1304     [(62, 8), (64, 8), tenor_two_musicmaker_one],
1305     [(64, 8), (66, 8), tenor_two_musicmaker_one],
1306     [(66, 8), (68, 8), tenor_two_musicmaker_one],
1307     [(68, 8), (70, 8), tenor_two_musicmaker_two],
1308     [(70, 8), (72, 8), tenor_two_musicmaker_two],
1309
1310     [(74, 8), (76, 8), tenor_two_musicmaker_one],
1311
1312     [(78, 8), (80, 8), tenor_two_musicmaker_two],
1313     [(80, 8), (82, 8), tenor_two_musicmaker_two],
1314     [(82, 8), (84, 8), tenor_two_musicmaker_one],
1315
1316     [(88, 8), (90, 8), tenor_two_musicmaker_two],
1317     [(90, 8), (92, 8), tenor_two_musicmaker_one],
1318
1319     [(94, 8), (96, 8), tenor_two_musicmaker_two],
1320     [(96, 8), (98, 8), tenor_two_musicmaker_two],
1321     [(98, 8), (100, 8), tenor_two_musicmaker_two],
1322
1323     [(102, 8), (104, 8), tenor_two_musicmaker_one],
1324
1325     [(106, 8), (108, 8), tenor_two_musicmaker_two],
1326     [(108, 8), (110, 8), tenor_two_musicmaker_one],
1327
1328     [(112, 8), (114, 8), tenor_two_musicmaker_two],
1329     [(114, 8), (116, 8), tenor_two_musicmaker_two],
1330
1331     [(118, 8), (120, 8), tenor_two_musicmaker_one],
1332 ]
1333 ])
1334
1335 voice_13_timestrap_list = abjad.TimespanList([
1336     abjad.AnnotatedTimespan(
1337         start_offset=start_offset,
1338         stop_offset=stop_offset,
1339         annotation=MusicSpecifier(
1340             music_maker=music_maker,
1341             voice_name='Voice 13',
1342         ),
1343     )
1344     for start_offset, stop_offset, music_maker in [
1345         [(0, 8), (2, 8), tenor_three_musicmaker_one],
1346         [(2, 8), (4, 8), tenor_three_musicmaker_one],
1347         [(4, 8), (6, 8), tenor_three_musicmaker_one],
1348
1349         [(16, 8), (18, 8), tenor_three_musicmaker_one],
1350         [(18, 8), (20, 8), tenor_three_musicmaker_one],
1351
1352         [(22, 8), (24, 8), tenor_three_musicmaker_one],
1353
1354         [(32, 8), (34, 8), tenor_three_musicmaker_one],
1355
1356         [(38, 8), (40, 8), tenor_three_musicmaker_two],
1357         [(40, 8), (42, 8), tenor_three_musicmaker_two],
1358         [(42, 8), (44, 8), tenor_three_musicmaker_two],
1359         [(44, 8), (46, 8), tenor_three_musicmaker_two],
1360
1361         [(58, 8), (60, 8), tenor_three_musicmaker_two],

```

```

1362     [(60, 8), (62, 8), tenor_three_musicmaker_two],
1363
1364     [(64, 8), (66, 8), tenor_three_musicmaker_one],
1365     [(66, 8), (68, 8), tenor_three_musicmaker_two],
1366
1367     [(76, 8), (78, 8), tenor_three_musicmaker_one],
1368     [(78, 8), (80, 8), tenor_three_musicmaker_one],
1369
1370     [(92, 8), (94, 8), tenor_three_musicmaker_two],
1371
1372     [(96, 8), (98, 8), tenor_three_musicmaker_one],
1373     [(98, 8), (100, 8), tenor_three_musicmaker_one],
1374     [(100, 8), (102, 8), tenor_three_musicmaker_one],
1375     [(102, 8), (104, 8), tenor_three_musicmaker_one],
1376
1377     [(116, 8), (118, 8), tenor_three_musicmaker_two],
1378     [(118, 8), (120, 8), tenor_three_musicmaker_two],
1379 ]
1380 ])
1381
1382 voice_14_timestrap_list = abjad.TimespanList([
1383     abjad.AnnotatedTimespan(
1384         start_offset=start_offset,
1385         stop_offset=stop_offset,
1386         annotation=MusicSpecifier(
1387             music_maker=music_maker,
1388             voice_name='Voice 14',
1389         ),
1390     )
1391     for start_offset, stop_offset, music_maker in [
1392         [(0, 8), (2, 8), tenor_four_musicmaker_one],
1393         [(2, 8), (4, 8), tenor_four_musicmaker_two],
1394
1395         [(6, 8), (8, 8), tenor_four_musicmaker_two],
1396
1397         [(10, 8), (12, 8), tenor_four_musicmaker_two],
1398         [(12, 8), (14, 8), tenor_four_musicmaker_one],
1399         [(14, 8), (16, 8), tenor_four_musicmaker_two],
1400         [(16, 8), (18, 8), tenor_four_musicmaker_one],
1401         [(18, 8), (20, 8), tenor_four_musicmaker_two],
1402
1403         [(22, 8), (24, 8), tenor_four_musicmaker_one],
1404
1405         [(28, 8), (30, 8), tenor_four_musicmaker_one],
1406         [(30, 8), (32, 8), tenor_four_musicmaker_two],
1407         [(32, 8), (34, 8), tenor_four_musicmaker_one],
1408         [(34, 8), (36, 8), tenor_four_musicmaker_one],
1409
1410         [(38, 8), (40, 8), tenor_four_musicmaker_two],
1411         [(40, 8), (42, 8), tenor_four_musicmaker_one],
1412         [(42, 8), (44, 8), tenor_four_musicmaker_one],
1413         [(44, 8), (46, 8), tenor_four_musicmaker_one],
1414
1415         [(48, 8), (50, 8), tenor_four_musicmaker_two],
1416         [(50, 8), (52, 8), tenor_four_musicmaker_two],
1417         [(52, 8), (54, 8), tenor_four_musicmaker_one],
1418         [(54, 8), (56, 8), tenor_four_musicmaker_one],
1419         [(56, 8), (58, 8), tenor_four_musicmaker_one],
1420

```

```

1421     [(60, 8), (62, 8), tenor_four_musicmaker_two],
1422     [(62, 8), (64, 8), tenor_four_musicmaker_two],
1423     [(64, 8), (66, 8), tenor_four_musicmaker_one],
1424     [(66, 8), (68, 8), tenor_four_musicmaker_one],
1425
1426     [(70, 8), (72, 8), tenor_four_musicmaker_two],
1427     [(72, 8), (74, 8), tenor_four_musicmaker_two],
1428     [(74, 8), (76, 8), tenor_four_musicmaker_two],
1429     [(76, 8), (78, 8), tenor_four_musicmaker_one],
1430
1431     [(80, 8), (82, 8), tenor_four_musicmaker_two],
1432     [(82, 8), (84, 8), tenor_four_musicmaker_one],
1433
1434     [(92, 8), (94, 8), tenor_four_musicmaker_two],
1435     [(94, 8), (96, 8), tenor_four_musicmaker_one],
1436     [(96, 8), (98, 8), tenor_four_musicmaker_one],
1437     [(98, 8), (100, 8), tenor_four_musicmaker_two],
1438     [(100, 8), (102, 8), tenor_four_musicmaker_one],
1439     [(102, 8), (104, 8), tenor_four_musicmaker_two],
1440
1441     [(106, 8), (108, 8), tenor_four_musicmaker_one],
1442     [(108, 8), (110, 8), tenor_four_musicmaker_one],
1443
1444     [(112, 8), (114, 8), tenor_four_musicmaker_one],
1445     [(114, 8), (116, 8), tenor_four_musicmaker_two],
1446     [(116, 8), (118, 8), tenor_four_musicmaker_one],
1447     [(118, 8), (120, 8), tenor_four_musicmaker_one],
1448 ]
1449 ])
1450
1451 voice_15_timespan_list = abjad.TimespanList([
1452     abjad.AnnotatedTimespan(
1453         start_offset=start_offset,
1454         stop_offset=stop_offset,
1455         annotation=MusicSpecifier(
1456             music_maker=music_maker,
1457             voice_name='Voice 15',
1458         ),
1459     )
1460     for start_offset, stop_offset, music_maker in [
1461         [(0, 8), (2, 8), tenor_five_musicmaker_two],
1462         [(2, 8), (4, 8), tenor_five_musicmaker_two],
1463
1464         [(6, 8), (8, 8), tenor_five_musicmaker_one],
1465         [(8, 8), (10, 8), tenor_five_musicmaker_two],
1466         [(10, 8), (12, 8), tenor_five_musicmaker_one],
1467
1468         [(14, 8), (16, 8), tenor_five_musicmaker_one],
1469
1470         [(18, 8), (20, 8), tenor_five_musicmaker_two],
1471
1472         [(22, 8), (24, 8), tenor_five_musicmaker_one],
1473         [(24, 8), (26, 8), tenor_five_musicmaker_two],
1474
1475         [(28, 8), (30, 8), tenor_five_musicmaker_one],
1476
1477         [(32, 8), (34, 8), tenor_five_musicmaker_two],
1478         [(34, 8), (36, 8), tenor_five_musicmaker_two],
1479

```

```

1480     [(40, 8), (42, 8), tenor_five_musicmaker_one],
1481     [(42, 8), (44, 8), tenor_five_musicmaker_two],
1482     [(44, 8), (46, 8), tenor_five_musicmaker_two],
1483
1484
1485     [(50, 8), (52, 8), tenor_five_musicmaker_two],
1486     [(52, 8), (54, 8), tenor_five_musicmaker_two],
1487
1488     [(58, 8), (60, 8), tenor_five_musicmaker_one],
1489     [(60, 8), (62, 8), tenor_five_musicmaker_one],
1490     [(62, 8), (64, 8), tenor_five_musicmaker_two],
1491
1492     [(66, 8), (68, 8), tenor_five_musicmaker_one],
1493
1494     [(70, 8), (72, 8), tenor_five_musicmaker_two],
1495
1496     [(74, 8), (76, 8), tenor_five_musicmaker_one],
1497     [(76, 8), (78, 8), tenor_five_musicmaker_two],
1498     [(78, 8), (80, 8), tenor_five_musicmaker_two],
1499
1500     [(82, 8), (84, 8), tenor_five_musicmaker_one],
1501     [(84, 8), (86, 8), tenor_five_musicmaker_two],
1502
1503     [(88, 8), (90, 8), tenor_five_musicmaker_one],
1504     [(90, 8), (92, 8), tenor_five_musicmaker_one],
1505
1506     [(94, 8), (96, 8), tenor_five_musicmaker_two],
1507     [(96, 8), (98, 8), tenor_five_musicmaker_one],
1508
1509     [(100, 8), (102, 8), tenor_five_musicmaker_two],
1510     [(102, 8), (104, 8), tenor_five_musicmaker_two],
1511     [(104, 8), (106, 8), tenor_five_musicmaker_one],
1512
1513     [(108, 8), (110, 8), tenor_five_musicmaker_one],
1514     [(110, 8), (112, 8), tenor_five_musicmaker_two],
1515     [(112, 8), (114, 8), tenor_five_musicmaker_two],
1516
1517     [(118, 8), (120, 8), tenor_five_musicmaker_two],
1518 ]
1519 ])
1520
1521 voice_16_timespan_list = abjad.TimespanList([
1522     abjad.AnnotatedTimespan(
1523         start_offset=start_offset,
1524         stop_offset=stop_offset,
1525         annotation=MusicSpecifier(
1526             music_maker=music_maker,
1527             voice_name='Voice 16',
1528         ),
1529     )
1530     for start_offset, stop_offset, music_maker in [
1531
1532         [(4, 8), (6, 8), baritone_one_musicmaker_one],
1533         [(6, 8), (8, 8), baritone_one_musicmaker_two],
1534
1535         [(10, 8), (12, 8), baritone_one_musicmaker_one],
1536         [(12, 8), (14, 8), baritone_one_musicmaker_two],
1537         [(14, 8), (16, 8), baritone_one_musicmaker_two],
1538         [(16, 8), (18, 8), baritone_one_musicmaker_one],

```

```

1539     [(20, 8), (22, 8), baritone_one_musicmaker_two],
1540     [(22, 8), (24, 8), baritone_one_musicmaker_one],
1541     [(26, 8), (28, 8), baritone_one_musicmaker_two],
1542
1543     [(32, 8), (34, 8), baritone_one_musicmaker_one],
1544
1545     [(36, 8), (38, 8), baritone_one_musicmaker_two],
1546     [(38, 8), (40, 8), baritone_one_musicmaker_one],
1547     [(40, 8), (42, 8), baritone_one_musicmaker_one],
1548
1549     [(44, 8), (46, 8), baritone_one_musicmaker_two],
1550     [(46, 8), (48, 8), baritone_one_musicmaker_two],
1551     [(48, 8), (50, 8), baritone_one_musicmaker_one],
1552
1553     [(52, 8), (54, 8), baritone_one_musicmaker_two],
1554     [(54, 8), (56, 8), baritone_one_musicmaker_two],
1555
1556     [(58, 8), (60, 8), baritone_one_musicmaker_one],
1557     [(60, 8), (62, 8), baritone_one_musicmaker_one],
1558
1559     [(64, 8), (66, 8), baritone_one_musicmaker_two],
1560
1561     [(68, 8), (70, 8), baritone_one_musicmaker_two],
1562     [(70, 8), (72, 8), baritone_one_musicmaker_two],
1563
1564     [(74, 8), (76, 8), baritone_one_musicmaker_one],
1565     [(76, 8), (78, 8), baritone_one_musicmaker_two],
1566
1567     [(86, 8), (88, 8), baritone_one_musicmaker_two],
1568     [(88, 8), (90, 8), baritone_one_musicmaker_two],
1569
1570     [(92, 8), (94, 8), baritone_one_musicmaker_one],
1571     [(94, 8), (96, 8), baritone_one_musicmaker_one],
1572     [(96, 8), (98, 8), baritone_one_musicmaker_two],
1573
1574     [(100, 8), (102, 8), baritone_one_musicmaker_one],
1575     [(102, 8), (104, 8), baritone_one_musicmaker_two],
1576
1577     [(106, 8), (108, 8), baritone_one_musicmaker_two],
1578
1579     [(110, 8), (112, 8), baritone_one_musicmaker_one],
1580     [(112, 8), (114, 8), baritone_one_musicmaker_two],
1581
1582 ]
1583 ])
1584
1585 voice_17_timespan_list = abjad.TimespanList([
1586     abjad.AnnotatedTimespan(
1587         start_offset=start_offset,
1588         stop_offset=stop_offset,
1589         annotation=MusicSpecifier(
1590             music_maker=music_maker,
1591             voice_name='Voice 17',
1592         ),
1593     )
1594     for start_offset, stop_offset, music_maker in [
1595         [(1, 8), (2, 8), baritone_two_musicmaker_one],
1596         [(3, 8), (4, 8), baritone_two_musicmaker_one],
1597     ]
]

```

```

1598     [(7, 8), (8, 8), baritone_two_musicmaker_one],
1599     [(9, 8), (10, 8), baritone_two_musicmaker_two],
1600
1601     [(13, 8), (14, 8), baritone_two_musicmaker_one],
1602     [(15, 8), (16, 8), baritone_two_musicmaker_two],
1603     [(17, 8), (18, 8), baritone_two_musicmaker_one],
1604
1605     [(22, 8), (24, 8), baritone_two_musicmaker_two],
1606     [(24, 8), (26, 8), baritone_two_musicmaker_one],
1607     [(26, 8), (28, 8), baritone_two_musicmaker_two],
1608
1609     [(30, 8), (32, 8), baritone_two_musicmaker_one],
1610
1611     [(35, 8), (36, 8), baritone_two_musicmaker_one],
1612     [(37, 8), (38, 8), baritone_two_musicmaker_one],
1613     [(39, 8), (40, 8), baritone_two_musicmaker_two],
1614
1615     [(42, 8), (44, 8), baritone_two_musicmaker_one],
1616     [(44, 8), (46, 8), baritone_two_musicmaker_two],
1617     [(46, 8), (48, 8), baritone_two_musicmaker_two],
1618     [(48, 8), (50, 8), baritone_two_musicmaker_two],
1619
1620     [(52, 8), (54, 8), baritone_two_musicmaker_one],
1621     [(54, 8), (56, 8), baritone_two_musicmaker_two],
1622     [(56, 8), (58, 8), baritone_two_musicmaker_two],
1623     [(58, 8), (60, 8), baritone_two_musicmaker_one],
1624
1625     [(63, 8), (64, 8), baritone_two_musicmaker_two],
1626     [(65, 8), (66, 8), baritone_two_musicmaker_one],
1627     [(67, 8), (68, 8), baritone_two_musicmaker_two],
1628
1629     [(71, 8), (72, 8), baritone_two_musicmaker_one],
1630     [(73, 8), (74, 8), baritone_two_musicmaker_two],
1631     [(75, 8), (76, 8), baritone_two_musicmaker_two],
1632
1633     [(78, 8), (80, 8), baritone_two_musicmaker_one],
1634     [(80, 8), (82, 8), baritone_two_musicmaker_one],
1635     [(82, 8), (84, 8), baritone_two_musicmaker_two],
1636
1637     [(86, 8), (88, 8), baritone_two_musicmaker_one],
1638     [(88, 8), (90, 8), baritone_two_musicmaker_one],
1639
1640     [(92, 8), (94, 8), baritone_two_musicmaker_two],
1641     [(94, 8), (96, 8), baritone_two_musicmaker_one],
1642     [(96, 8), (98, 8), baritone_two_musicmaker_two],
1643
1644     [(100, 8), (102, 8), baritone_two_musicmaker_one],
1645     [(102, 8), (104, 8), baritone_two_musicmaker_two],
1646     [(104, 8), (106, 8), baritone_two_musicmaker_one],
1647
1648     [(108, 8), (110, 8), baritone_two_musicmaker_two],
1649
1650     [(112, 8), (114, 8), baritone_two_musicmaker_one],
1651     [(114, 8), (116, 8), baritone_two_musicmaker_one],
1652     [(116, 8), (118, 8), baritone_two_musicmaker_two],
1653     [(118, 8), (120, 8), baritone_two_musicmaker_two],
1654 ]
1655 ])
1656

```

```
1657 voice_18_timespan_list = abjad.TimespanList([
1658     abjad.AnnotatedTimespan(
1659         start_offset=start_offset,
1660         stop_offset=stop_offset,
1661         annotation=MusicSpecifier(
1662             music_maker=music_maker,
1663             voice_name='Voice 18',
1664         ),
1665     )
1666     for start_offset, stop_offset, music_maker in [
1667         [(0, 8), (2, 8), baritone_three_musicmaker_two],
1668         [(2, 8), (4, 8), baritone_three_musicmaker_one],
1669         [(6, 8), (8, 8), baritone_three_musicmaker_one],
1670         [(10, 8), (12, 8), baritone_three_musicmaker_one],
1671         [(12, 8), (14, 8), baritone_three_musicmaker_two],
1672         [(14, 8), (16, 8), baritone_three_musicmaker_two],
1673         [(20, 8), (22, 8), baritone_three_musicmaker_one],
1674         [(22, 8), (24, 8), baritone_three_musicmaker_two],
1675         [(24, 8), (26, 8), baritone_three_musicmaker_one],
1676         [(32, 8), (34, 8), baritone_three_musicmaker_two],
1677         [(34, 8), (36, 8), baritone_three_musicmaker_two],
1678         [(38, 8), (40, 8), baritone_three_musicmaker_two],
1679         [(42, 8), (44, 8), baritone_three_musicmaker_one],
1680         [(44, 8), (46, 8), baritone_three_musicmaker_two],
1681         [(48, 8), (50, 8), baritone_three_musicmaker_two],
1682         [(50, 8), (52, 8), baritone_three_musicmaker_two],
1683         [(52, 8), (54, 8), baritone_three_musicmaker_one],
1684         [(56, 8), (58, 8), baritone_three_musicmaker_one],
1685         [(60, 8), (62, 8), baritone_three_musicmaker_two],
1686         [(62, 8), (64, 8), baritone_three_musicmaker_two],
1687         [(64, 8), (66, 8), baritone_three_musicmaker_one],
1688         [(66, 8), (68, 8), baritone_three_musicmaker_one],
1689         [(68, 8), (70, 8), baritone_three_musicmaker_two],
1690         [(72, 8), (74, 8), baritone_three_musicmaker_two],
1691         [(74, 8), (76, 8), baritone_three_musicmaker_one],
1692         [(78, 8), (80, 8), baritone_three_musicmaker_two],
1693         [(80, 8), (82, 8), baritone_three_musicmaker_two],
1694         [(82, 8), (84, 8), baritone_three_musicmaker_one],
1695         [(84, 8), (86, 8), baritone_three_musicmaker_one],
1696         [(86, 8), (88, 8), baritone_three_musicmaker_one],
1697         [(90, 8), (92, 8), baritone_three_musicmaker_two],
1698         [(94, 8), (96, 8), baritone_three_musicmaker_one],
1699         [(96, 8), (98, 8), baritone_three_musicmaker_one],
1700         [(98, 8), (100, 8), baritone_three_musicmaker_two],
1701         [(100, 8), (102, 8), baritone_three_musicmaker_two],
1702     ]
```

```

1716     [(106, 8), (108, 8), baritone_three_musicmaker_one],
1717     [(108, 8), (110, 8), baritone_three_musicmaker_two],
1718
1719     [(112, 8), (114, 8), baritone_three_musicmaker_one],
1720     [(114, 8), (116, 8), baritone_three_musicmaker_two],
1721     [(116, 8), (118, 8), baritone_three_musicmaker_two],
1722     [(118, 8), (120, 8), baritone_three_musicmaker_one],
1723 ]
1724 ])
1725
1726 voice_19_timespan_list = abjad.TimespanList([
1727     abjad.AnnotatedTimespan(
1728         start_offset=start_offset,
1729         stop_offset=stop_offset,
1730         annotation=MusicSpecifier(
1731             music_maker=music_maker,
1732             voice_name='Voice 19',
1733         ),
1734     )
1735     for start_offset, stop_offset, music_maker in [
1736         [(0, 8), (2, 8), bass_one_musicmaker_one],
1737
1738         [(4, 8), (6, 8), bass_one_musicmaker_one],
1739         [(6, 8), (8, 8), bass_one_musicmaker_two],
1740         [(8, 8), (10, 8), bass_one_musicmaker_one],
1741         [(10, 8), (12, 8), bass_one_musicmaker_one],
1742         [(12, 8), (14, 8), bass_one_musicmaker_one],
1743
1744         [(16, 8), (18, 8), bass_one_musicmaker_one],
1745         [(18, 8), (20, 8), bass_one_musicmaker_one],
1746         [(20, 8), (22, 8), bass_one_musicmaker_one],
1747         [(22, 8), (24, 8), bass_one_musicmaker_two],
1748
1749         [(26, 8), (28, 8), bass_one_musicmaker_one],
1750         [(28, 8), (30, 8), bass_one_musicmaker_one],
1751         [(30, 8), (32, 8), bass_one_musicmaker_two],
1752
1753         [(34, 8), (36, 8), bass_one_musicmaker_one],
1754         [(36, 8), (38, 8), bass_one_musicmaker_one],
1755
1756         [(50, 8), (52, 8), bass_one_musicmaker_two],
1757         [(52, 8), (54, 8), bass_one_musicmaker_one],
1758         [(54, 8), (56, 8), bass_one_musicmaker_one],
1759         [(56, 8), (58, 8), bass_one_musicmaker_two],
1760
1761         [(60, 8), (62, 8), bass_one_musicmaker_one],
1762         [(62, 8), (64, 8), bass_one_musicmaker_one],
1763         [(64, 8), (68, 8), bass_one_musicmaker_two],
1764         [(68, 8), (70, 8), bass_one_musicmaker_one],
1765
1766         [(72, 8), (74, 8), bass_one_musicmaker_one],
1767         [(74, 8), (76, 8), bass_one_musicmaker_two],
1768         [(76, 8), (78, 8), bass_one_musicmaker_one],
1769         [(78, 8), (80, 8), bass_one_musicmaker_one],
1770         [(80, 8), (82, 8), bass_one_musicmaker_one],
1771         [(82, 8), (84, 8), bass_one_musicmaker_two],
1772
1773         [(88, 8), (90, 8), bass_one_musicmaker_one],
1774         [(90, 8), (92, 8), bass_one_musicmaker_two],

```

```

1775     [(92, 8), (94, 8), bass_one_musicmaker_one],
1776     [(94, 8), (96, 8), bass_one_musicmaker_one],
1777     [(96, 8), (98, 8), bass_one_musicmaker_one],
1778
1779     [(100, 8), (102, 8), bass_one_musicmaker_one],
1780     [(102, 8), (104, 8), bass_one_musicmaker_one],
1781     [(104, 8), (106, 8), bass_one_musicmaker_one],
1782     [(106, 8), (108, 8), bass_one_musicmaker_two],
1783     [(108, 8), (110, 8), bass_one_musicmaker_one],
1784     [(110, 8), (112, 8), bass_one_musicmaker_one],
1785
1786     [(114, 8), (116, 8), bass_one_musicmaker_two],
1787     [(116, 8), (118, 8), bass_one_musicmaker_one],
1788     [(118, 8), (120, 8), bass_one_musicmaker_one],
1789 ]
1790 ])
1791
1792 voice_20_timespan_list = abjad.TimespanList([
1793     abjad.AnnotatedTimespan(
1794         start_offset=start_offset,
1795         stop_offset=stop_offset,
1796         annotation=MusicSpecifier(
1797             music_maker=music_maker,
1798             voice_name='Voice 20',
1799         ),
1800     )
1801     for start_offset, stop_offset, music_maker in [
1802         [(0, 8), (2, 8), bass_two_musicmaker_two],
1803         [(2, 8), (4, 8), bass_two_musicmaker_two],
1804         [(4, 8), (6, 8), bass_two_musicmaker_two],
1805         [(6, 8), (8, 8), bass_two_musicmaker_two],
1806
1807         [(16, 8), (18, 8), bass_two_musicmaker_two],
1808
1809         [(20, 8), (22, 8), bass_two_musicmaker_two],
1810         [(22, 8), (24, 8), bass_two_musicmaker_one],
1811
1812         [(26, 8), (28, 8), bass_two_musicmaker_one],
1813
1814         [(40, 8), (42, 8), bass_two_musicmaker_one],
1815         [(42, 8), (44, 8), bass_two_musicmaker_two],
1816         [(44, 8), (46, 8), bass_two_musicmaker_two],
1817         [(46, 8), (48, 8), bass_two_musicmaker_two],
1818
1819         [(50, 8), (52, 8), bass_two_musicmaker_one],
1820
1821         [(64, 8), (68, 8), bass_two_musicmaker_two],
1822         [(68, 8), (70, 8), bass_two_musicmaker_two],
1823         [(70, 8), (72, 8), bass_two_musicmaker_one],
1824
1825         [(74, 8), (76, 8), bass_two_musicmaker_one],
1826         [(76, 8), (78, 8), bass_two_musicmaker_two],
1827         [(78, 8), (80, 8), bass_two_musicmaker_two],
1828
1829         [(88, 8), (90, 8), bass_two_musicmaker_one],
1830         [(90, 8), (92, 8), bass_two_musicmaker_two],
1831
1832         [(94, 8), (96, 8), bass_two_musicmaker_two],
1833         [(96, 8), (98, 8), bass_two_musicmaker_two],

```

```

1834     [(98, 8), (100, 8), bass_two_musicmaker_one],
1835     [(100, 8), (102, 8), bass_two_musicmaker_one],
1836
1837     [(108, 8), (110, 8), bass_two_musicmaker_two],
1838     [(110, 8), (112, 8), bass_two_musicmaker_two],
1839     [(112, 8), (114, 8), bass_two_musicmaker_one],
1840     [(114, 8), (116, 8), bass_two_musicmaker_one],
1841     [(116, 8), (118, 8), bass_two_musicmaker_one],
1842 ]
1843 ])
1844
1845 voice_21_timestspan_list = abjad.TimespanList([
1846     abjad.AnnotatedTimespan(
1847         start_offset=start_offset,
1848         stop_offset=stop_offset,
1849         annotation=MusicSpecifier(
1850             music_maker=music_maker,
1851             voice_name='Voice 21',
1852         ),
1853     )
1854     for start_offset, stop_offset, music_maker in [
1855         [(1, 8), (2, 8), contrabass_musicmaker_two],
1856         [(3, 8), (4, 8), contrabass_musicmaker_two],
1857
1858         [(7, 8), (8, 8), contrabass_musicmaker_one],
1859         [(9, 8), (10, 8), contrabass_musicmaker_one],
1860         [(11, 8), (12, 8), contrabass_musicmaker_two],
1861
1862         [(15, 8), (16, 8), contrabass_musicmaker_one],
1863         [(17, 8), (18, 8), contrabass_musicmaker_one],
1864         [(19, 8), (20, 8), contrabass_musicmaker_one],
1865
1866         [(23, 8), (24, 8), contrabass_musicmaker_two],
1867         [(25, 8), (26, 8), contrabass_musicmaker_one],
1868         [(27, 8), (28, 8), contrabass_musicmaker_one],
1869         [(29, 8), (30, 8), contrabass_musicmaker_one],
1870
1871         [(33, 8), (34, 8), contrabass_musicmaker_two],
1872         [(35, 8), (36, 8), contrabass_musicmaker_one],
1873         [(37, 8), (38, 8), contrabass_musicmaker_one],
1874         [(39, 8), (40, 8), contrabass_musicmaker_one],
1875         [(41, 8), (42, 8), contrabass_musicmaker_two],
1876
1877
1878         [(46, 8), (48, 8), contrabass_musicmaker_one],
1879         [(48, 8), (50, 8), contrabass_musicmaker_one],
1880         [(50, 8), (52, 8), contrabass_musicmaker_two],
1881
1882         [(54, 8), (56, 8), contrabass_musicmaker_one],
1883         [(56, 8), (58, 8), contrabass_musicmaker_one],
1884         [(58, 8), (60, 8), contrabass_musicmaker_one],
1885
1886         [(63, 8), (64, 8), contrabass_musicmaker_two],
1887         [(65, 8), (68, 8), contrabass_musicmaker_one],
1888
1889         [(70, 8), (72, 8), contrabass_musicmaker_one],
1890         [(72, 8), (74, 8), contrabass_musicmaker_two],
1891         [(74, 8), (76, 8), contrabass_musicmaker_two],
1892         [(76, 8), (78, 8), contrabass_musicmaker_one],

```

```

1893     [(78, 8), (80, 8), contrabass_musicmaker_one],
1894
1895     [(82, 8), (84, 8), contrabass_musicmaker_two],
1896
1897     [(90, 8), (92, 8), contrabass_musicmaker_one],
1898     [(92, 8), (94, 8), contrabass_musicmaker_two],
1899     [(94, 8), (96, 8), contrabass_musicmaker_two],
1900
1901     [(98, 8), (100, 8), contrabass_musicmaker_one],
1902     [(100, 8), (102, 8), contrabass_musicmaker_one],
1903
1904     [(104, 8), (106, 8), contrabass_musicmaker_two],
1905     [(106, 8), (108, 8), contrabass_musicmaker_one],
1906
1907     [(111, 8), (112, 8), contrabass_musicmaker_one],
1908     [(113, 8), (114, 8), contrabass_musicmaker_two],
1909
1910     [(117, 8), (118, 8), contrabass_musicmaker_one],
1911 ]
1912 ])
1913
1914 all_timespan_lists = {
1915     'Voice 1': voice_1_timespan_list,
1916     'Voice 2': voice_2_timespan_list,
1917     'Voice 3': voice_3_timespan_list,
1918     'Voice 4': voice_4_timespan_list,
1919     'Voice 5': voice_5_timespan_list,
1920     'Voice 6': voice_6_timespan_list,
1921     'Voice 7': voice_7_timespan_list,
1922     'Voice 8': voice_8_timespan_list,
1923     'Voice 9': voice_9_timespan_list,
1924     'Voice 10': voice_10_timespan_list,
1925     'Voice 11': voice_11_timespan_list,
1926     'Voice 12': voice_12_timespan_list,
1927     'Voice 13': voice_13_timespan_list,
1928     'Voice 14': voice_14_timespan_list,
1929     'Voice 15': voice_15_timespan_list,
1930     'Voice 16': voice_16_timespan_list,
1931     'Voice 17': voice_17_timespan_list,
1932     'Voice 18': voice_18_timespan_list,
1933     'Voice 19': voice_19_timespan_list,
1934     'Voice 20': voice_20_timespan_list,
1935     'Voice 21': voice_21_timespan_list,
1936 }
1937
1938 global_timespan = abjad.Timespan(
1939     start_offset=0,
1940     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
1941 )
1942
1943 for voice_name, timespan_list in all_timespan_lists.items():
1944     silences = abjad.TimespanList([global_timespan])
1945     silences.extend(timespan_list)
1946     silences.sort()
1947     silences.compute_logical_xor()
1948     for silence_timespan in silences:
1949         timespan_list.append(
1950             abjad.AnnotatedTimespan(
1951                 start_offset=silence_timespan.start_offset,

```

```

1952         stop_offset=silence_timespan.stop_offset,
1953         annotation=MusicSpecifier(
1954             music_maker=None,
1955             voice_name=voice_name,
1956         ),
1957     ),
1958 )
1959 timespan_list.sort()
1960
1961 for voice_name, timespan_list in all_timespan_lists.items():
1962     shards = timespan_list.split_at_offsets(bounds)
1963     split_timespan_list = abjad.TimespanList()
1964     for shard in shards:
1965         split_timespan_list.extend(shard)
1966     split_timespan_list.sort()
1967     all_timespan_lists[voice_name] = timespan_list
1968
1969 score = abjad.Score([
1970     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
1971     abjad.StaffGroup(
1972         [
1973             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
1974                 lilypond_type='Staff',),
1975                 abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
1976                 lilypond_type='Staff',),
1977                 abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
1978                 lilypond_type='Staff',),
1979                 abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
1980                 lilypond_type='Staff',),
1981                 abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
1982                 lilypond_type='Staff',),
1983                 abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
1984                 lilypond_type='Staff',),
1985                 abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
1986                 lilypond_type='Staff',),
1987                 abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
1988                 lilypond_type='Staff',),
1989                 abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
1990                 lilypond_type='Staff',),
1991                 abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
1992                 lilypond_type='Staff',),
1993                 abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
1994                 lilypond_type='Staff',),
1995                 abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
1996                 lilypond_type='Staff',),
1997                 abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
1998                 lilypond_type='Staff',),
1999                 abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
2000                 lilypond_type='Staff',),
2001                 abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
2002                 lilypond_type='Staff',),
2003                 abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
2004                 lilypond_type='Staff',),
2005                 abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
2006                 lilypond_type='Staff',),
2007                 abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
2008                 lilypond_type='Staff',),
2009                 abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
2010                 lilypond_type='Staff',),
2011             ]
2012         )
2013     )
2014 
```

```

1992         abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
1993             lilypond_type='Staff',),
1994                 abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
1995                     lilypond_type='Staff',),
1996                         ],
1997                         name='Staff Group',
1998 )
1999 ]
2000
2001     for time_signature in time_signatures:
2002         skip = abjad.Skip(1, multiplier=(time_signature))
2003         abjad.attach(time_signature, skip)
2004         score['Global Context'].append(skip)
2005
2006     print('Making containers ...')
2007
2008     def make_container(music_maker, durations):
2009         selections = music_maker(durations)
2010         container = abjad.Container([])
2011         container.extend(selections)
2012         return container
2013
2014     def key_function(timespan):
2015         return timespan.annotation.music_maker or silence_maker
2016
2017     for voice_name, timespan_list in all_timespan_lists.items():
2018         for music_maker, grouper in itertools.groupby(
2019             timespan_list,
2020             key=key_function,
2021         ):
2022             durations = [timespan.duration for timespan in grouper]
2023             container = make_container(music_maker, durations)
2024             voice = score[voice_name]
2025             voice.append(container)
2026
2027     print('Splitting and rewriting ...')
2028     for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
2029         for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
2030             time_signature = time_signatures[i]
2031             abjad.mutate(shard).rewrite_meter(time_signature)
2032
2033     print('Beaming runs ...')
2034     for voice in abjad.select(score).components(abjad.Voice):
2035         for run in abjad.select(voice).runs():
2036             specifier = abjadext.rmakers.BeamSpecifier(
2037                 beam_each_division=False,
2038             )
2039             specifier(run)
2040             abjad.beam(voice[:], beam_lone_notes=False, beam_rests=False,)
2041
2042     print('Stopping Hairpins ...')
2043     for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2044         for rest in abjad.iterate(staff).components(abjad.Rest):
2045             previous_leaf = abjad.inspect(rest).leaf(-1)
2046             if isinstance(previous_leaf, abjad.Note):
2047                 abjad.attach(abjad.StopHairpin(), rest)
2048             elif isinstance(previous_leaf, abjad.Chord):
2049                 abjad.attach(abjad.StopHairpin(), rest)
2050             elif isinstance(previous_leaf, abjad.Rest):
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
41
```

```

2049     pass
2050
2051 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2052     first_leaf = abjad.select(staff).leaves()[0]
2053     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
2054     abjad.attach(stop, first_leaf)
2055
2056 print('Adding attachments ...')
2057 bar_line = abjad.BarLine('||')
2058 metro = abjad.MetronomeMark((1, 4), 108)
2059 markup = abjad.Markup(r'\bold { D }')
2060 mark = abjad.RehearsalMark(markup=markup)
2061
2062 instruments = cyc([
2063     abjad.SopranoSaxophone(),
2064     abjad.SopranoSaxophone(),
2065     abjad.SopranoSaxophone(),
2066     abjad.SopranoSaxophone(),
2067     abjad.AltoSaxophone(),
2068     abjad.AltoSaxophone(),
2069     abjad.AltoSaxophone(),
2070     abjad.AltoSaxophone(),
2071     abjad.AltoSaxophone(),
2072     abjad.AltoSaxophone(),
2073     abjad.TenorSaxophone(),
2074     abjad.TenorSaxophone(),
2075     abjad.TenorSaxophone(),
2076     abjad.TenorSaxophone(),
2077     abjad.TenorSaxophone(),
2078     abjad.BaritoneSaxophone(),
2079     abjad.BaritoneSaxophone(),
2080     abjad.BaritoneSaxophone(),
2081     abjad.BassSaxophone(),
2082     abjad.BassSaxophone(),
2083     abjad.ContrabassSaxophone(),
2084 ])
2085
2086 abbreviations = cyc([
2087     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
2088     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
2089     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
2090     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
2091     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
2092     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
2093     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
2094     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
2095     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
2096     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
2097     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
2098     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
2099     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
2100     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
2101     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
2102     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
2103     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
2104     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
2105     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
2106     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
2107     abjad.MarginMarkup(markup=abjad.Markup('cbs.')),
```

```

2108 ])
2109
2110 names = cyc([
2111     abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
2112     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
2113     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
2114     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
2115     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
2116     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
2117     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
2118     abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
2119     abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
2120     abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
2121     abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
2122     abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
2123     abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
2124     abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
2125     abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
2126     abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
2127     abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
2128     abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
2129     abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
2130     abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
2131     abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
2132 ])
2133
2134 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2135     leaf1 = abjad.select(staff).leaves()[0]
2136     abjad.attach(next(instruments), leaf1)
2137     abjad.attach(next(abbreviations), leaf1)
2138     abjad.attach(next(names), leaf1)
2139
2140 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
2141     leaf1 = abjad.select(staff).leaves()[0]
2142     last_leaf = abjad.select(staff).leaves()[-1]
2143     abjad.attach(metro, leaf1)
2144     abjad.attach(bar_line, last_leaf)
2145
2146 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
2147     leaf1 = abjad.select(staff).leaves()[0]
2148     abjad.attach(mark, leaf1)
2149
2150 score_file = abjad.LilyPondFile.new(
2151     score,
2152     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2153     _stylesheets/abjad.ily'],
2154 )
2155
2156 abjad.SegmentMaker.comment_measure_numbers(score)
2157 ######
2158 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_D'
2159 pdf_path = f'{directory}/Section_D.pdf'
2160 path = pathlib.Path('Section_D.pdf')
2161 if path.exists():
2162     print(f'Removing {pdf_path} ...')
2163     path.unlink()
2164 time_1 = time.time()
2165 print(f'Persisting {pdf_path} ...')

```

```

2166 result = abjad.persist(score_file).as_pdf(pdf_path)
2167 print(result[0])
2168 print(result[1])
2169 print(result[2])
2170 success = result[3]
2171 if success is False:
2172     print('LilyPond failed!')
2173 time_2 = time.time()
2174 total_time = time_2 - time_1
2175 print(f'Total time: {total_time} seconds')
2176 if path.exists():
2177     print(f'Opening {pdf_path} ...')
2178     os.system(f'open {pdf_path}')
2179 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_D/Section_D.ly').readlines()
2180 open('/Users/evansdsg2/Scores/guerrero/Build/Section_D.ly', 'w').writelines(
2181     score_lines[15:-1])
2182
2183 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
2184     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2185     signature_copy = abjad.mutate(signatures).copy()
2186     staff_copy = abjad.mutate(staff).copy()
2187     part = abjad.Score()
2188     part.insert(0, staff)
2189     part.insert(0, signature_copy)
2190     part_file = abjad.LilyPondFile.new(
2191         part,
2192         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/_stylesheets/abjad.ily'],
2193         )
2194     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano'
2195     pdf_path = f'{directory}/Section_D.pdf'
2196     path = pathlib.Path('Section_D.pdf')
2197     if path.exists():
2198         print(f'Removing {pdf_path} ...')
2199         path.unlink()
2200     time_1 = time.time()
2201     print(f'Persisting {pdf_path} ...')
2202     result = abjad.persist(part_file).as_pdf(pdf_path)
2203     print(result[0])
2204     print(result[1])
2205     print(result[2])
2206     success = result[3]
2207     if success is False:
2208         print('LilyPond failed!')
2209     time_2 = time.time()
2210     total_time = time_2 - time_1
2211     print(f'Total time: {total_time} seconds')
2212     if path.exists():
2213         print(f'Opening {pdf_path} ...')
2214         os.system(f'open {pdf_path}')
2215     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano'
2216     open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano/Section_D.ly',
2217     'w').writelines(part_lines[15:-1])
2218
2219 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
2220     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2221     signature_copy = abjad.mutate(signatures).copy()

```

```

2220 staff_copy = abjad.mutate(staff).copy()
2221 part = abjad.Score()
2222 part.insert(0, staff)
2223 part.insert(0, signature_copy)
2224 part_file = abjad.LilyPondFile.new(
2225     part,
2226     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2227 _stylesheets/abjad.ily'],
2228 )
2229 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1'
2230 pdf_path = f'{directory}/Section_D.pdf'
2231 path = pathlib.Path('Section_D.pdf')
2232 if path.exists():
2233     print(f'Removing {pdf_path} ...')
2234     path.unlink()
2235 time_1 = time.time()
2236 print(f'Persisting {pdf_path} ...')
2237 result = abjad.persist(part_file).as_pdf(pdf_path)
2238 print(result[0])
2239 print(result[1])
2240 print(result[2])
2241 success = result[3]
2242 if success is False:
2243     print('LilyPond failed!')
2244 time_2 = time.time()
2245 total_time = time_2 - time_1
2246 print(f'Total time: {total_time} seconds')
2247 if path.exists():
2248     print(f'Opening {pdf_path} ...')
2249     os.system(f'open {pdf_path}')
2250 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/
2251 Section_D.ly').readlines()
2252 open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/Section_D.ly',
2253 'w').writelines(part_lines[15:-1])
2254
2255 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
2256     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2257     signature_copy = abjad.mutate(signatures).copy()
2258     staff_copy = abjad.mutate(staff).copy()
2259     part = abjad.Score()
2260     part.insert(0, staff)
2261     part.insert(0, signature_copy)
2262     part_file = abjad.LilyPondFile.new(
2263         part,
2264         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2265 _stylesheets/abjad.ily'],
2266     )
2267     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
2268     pdf_path = f'{directory}/Section_D.pdf'
2269     path = pathlib.Path('Section_D.pdf')
2270     if path.exists():
2271         print(f'Removing {pdf_path} ...')
2272         path.unlink()
2273     time_1 = time.time()
2274     print(f'Persisting {pdf_path} ...')
2275     result = abjad.persist(part_file).as_pdf(pdf_path)
2276     print(result[0])
2277     print(result[1])
2278     print(result[2])

```

```

2275     success = result[3]
2276     if success is False:
2277         print('LilyPond failed!')
2278     time_2 = time.time()
2279     total_time = time_2 - time_1
2280     print(f'Total time: {total_time} seconds')
2281     if path.exists():
2282         print(f'Opening {pdf_path} ...')
2283         os.system(f'open {pdf_path}')
2284     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/
2285     Section_D.ly').readlines()
2286     open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/Section_D.ly',
2287     'w').writelines(part_lines[15:-1])
2288
2289 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
2290     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2291     signature_copy = abjad.mutate(signatures).copy()
2292     staff_copy = abjad.mutate(staff).copy()
2293     part = abjad.Score()
2294     part.insert(0, staff)
2295     part.insert(0, signature_copy)
2296     part_file = abjad.LilyPondFile.new(
2297         part,
2298         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2299         _stylesheets/abjad.ily'],
2300         )
2301     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3'
2302     pdf_path = f'{directory}/Section_D.pdf'
2303     path = pathlib.Path('Section_D.pdf')
2304     if path.exists():
2305         print(f'Removing {pdf_path} ...')
2306         path.unlink()
2307     time_1 = time.time()
2308     print(f'Persisting {pdf_path} ...')
2309     result = abjad.persist(part_file).as_pdf(pdf_path)
2310     print(result[0])
2311     print(result[1])
2312     print(result[2])
2313     success = result[3]
2314     if success is False:
2315         print('LilyPond failed!')
2316     time_2 = time.time()
2317     total_time = time_2 - time_1
2318     print(f'Total time: {total_time} seconds')
2319     if path.exists():
2320         print(f'Opening {pdf_path} ...')
2321         os.system(f'open {pdf_path}')
2322     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/
2323     Section_D.ly').readlines()
2324     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/Section_D.ly',
2325     'w').writelines(part_lines[15:-1])
2326
2327 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
2328     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2329     signature_copy = abjad.mutate(signatures).copy()
2330     staff_copy = abjad.mutate(staff).copy()
2331     part = abjad.Score()
2332     part.insert(0, staff)
2333     part.insert(0, signature_copy)

```

```

2329     part_file = abjad.LilyPondFile.new(
2330         part,
2331         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2332         _stylesheets/abjad.ily'],
2333         )
2334     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1'
2335     pdf_path = f'{directory}/Section_D.pdf'
2336     path = pathlib.Path('Section_D.pdf')
2337     if path.exists():
2338         print(f'Removing {pdf_path} ...')
2339         path.unlink()
2340     time_1 = time.time()
2341     print(f'Persisting {pdf_path} ...')
2342     result = abjad.persist(part_file).as_pdf(pdf_path)
2343     print(result[0])
2344     print(result[1])
2345     print(result[2])
2346     success = result[3]
2347     if success is False:
2348         print('LilyPond failed!')
2349     time_2 = time.time()
2350     total_time = time_2 - time_1
2351     print(f'Total time: {total_time} seconds')
2352     if path.exists():
2353         print(f'Opening {pdf_path} ...')
2354         os.system(f'open {pdf_path}')
2355     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/
2356     Section_D.ly').readlines()
2357     open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/Section_D.ly', 'w'
2358     ).writelines(part_lines[15:-1])
2359
2360 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
2361     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2362     signature_copy = abjad.mutate(signatures).copy()
2363     staff_copy = abjad.mutate(staff).copy()
2364     part = abjad.Score()
2365     part.insert(0, staff)
2366     part.insert(0, signature_copy)
2367     part_file = abjad.LilyPondFile.new(
2368         part,
2369         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2370         _stylesheets/abjad.ily'],
2371         )
2372     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
2373     pdf_path = f'{directory}/Section_D.pdf'
2374     path = pathlib.Path('Section_D.pdf')
2375     if path.exists():
2376         print(f'Removing {pdf_path} ...')
2377         path.unlink()
2378     time_1 = time.time()
2379     print(f'Persisting {pdf_path} ...')
2380     result = abjad.persist(part_file).as_pdf(pdf_path)
2381     print(result[0])
2382     print(result[1])
2383     print(result[2])
2384     success = result[3]
2385     if success is False:
2386         print('LilyPond failed!')
2387     time_2 = time.time()

```

```

2384     total_time = time_2 - time_1
2385     print(f'Total time: {total_time} seconds')
2386     if path.exists():
2387         print(f'Opening {pdf_path} ...')
2388         os.system(f'open {pdf_path}')
2389     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
2390     Section_D.ly').readlines()
2391     open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_D.ly', 'w'
2392     ).writelines(part_lines[15:-1])
2393
2394 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
2395     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2396     signature_copy = abjad.mutate(signatures).copy()
2397     staff_copy = abjad.mutate(staff).copy()
2398     part = abjad.Score()
2399     part.insert(0, staff)
2400     part.insert(0, signature_copy)
2401     part_file = abjad.LilyPondFile.new(
2402         part,
2403         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2404         _stylesheets/abjad.ily'],
2405         )
2406     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
2407     pdf_path = f'{directory}/Section_D.pdf'
2408     path = pathlib.Path('Section_D.pdf')
2409     if path.exists():
2410         print(f'Removing {pdf_path} ...')
2411         path.unlink()
2412     time_1 = time.time()
2413     print(f'Persisting {pdf_path} ...')
2414     result = abjad.persist(part_file).as_pdf(pdf_path)
2415     print(result[0])
2416     print(result[1])
2417     print(result[2])
2418     success = result[3]
2419     if success is False:
2420         print('LilyPond failed!')
2421     time_2 = time.time()
2422     total_time = time_2 - time_1
2423     print(f'Total time: {total_time} seconds')
2424     if path.exists():
2425         print(f'Opening {pdf_path} ...')
2426         os.system(f'open {pdf_path}')
2427     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/
2428     Section_D.ly').readlines()
2429     open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/Section_D.ly', 'w'
2430     ).writelines(part_lines[15:-1])
2431
2432 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
2433     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2434     signature_copy = abjad.mutate(signatures).copy()
2435     staff_copy = abjad.mutate(staff).copy()
2436     part = abjad.Score()
2437     part.insert(0, staff)
2438     part.insert(0, signature_copy)
2439     part_file = abjad.LilyPondFile.new(
2440         part,
2441         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2442         _stylesheets/abjad.ily'],

```

```

2437     )
2438     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
2439     pdf_path = f'{directory}/Section_D.pdf'
2440     path = pathlib.Path('Section_D.pdf')
2441     if path.exists():
2442         print(f'Removing {pdf_path} ...')
2443         path.unlink()
2444     time_1 = time.time()
2445     print(f'Persisting {pdf_path} ...')
2446     result = abjad.persist(part_file).as_pdf(pdf_path)
2447     print(result[0])
2448     print(result[1])
2449     print(result[2])
2450     success = result[3]
2451     if success is False:
2452         print('LilyPond failed!')
2453     time_2 = time.time()
2454     total_time = time_2 - time_1
2455     print(f'Total time: {total_time} seconds')
2456     if path.exists():
2457         print(f'Opening {pdf_path} ...')
2458         os.system(f'open {pdf_path}')
2459     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/
2460     Section_D.ly').readlines()
2461     open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/Section_D.ly', 'w'
2462     ).writelines(part_lines[15:-1])
2463
2464 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
2465     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2466     signature_copy = abjad.mutate(signatures).copy()
2467     staff_copy = abjad.mutate(staff).copy()
2468     part = abjad.Score()
2469     part.insert(0, staff)
2470     part.insert(0, signature_copy)
2471     part_file = abjad.LilyPondFile.new(
2472         part,
2473         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
2474         _stylesheets/abjad.ly'],
2475         )
2476     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
2477     pdf_path = f'{directory}/Section_D.pdf'
2478     path = pathlib.Path('Section_D.pdf')
2479     if path.exists():
2480         print(f'Removing {pdf_path} ...')
2481         path.unlink()
2482     time_1 = time.time()
2483     print(f'Persisting {pdf_path} ...')
2484     result = abjad.persist(part_file).as_pdf(pdf_path)
2485     print(result[0])
2486     print(result[1])
2487     print(result[2])
2488     success = result[3]
2489     if success is False:
2490         print('LilyPond failed!')
2491     time_2 = time.time()
2492     total_time = time_2 - time_1
2493     print(f'Total time: {total_time} seconds')
2494     if path.exists():
2495         print(f'Opening {pdf_path} ...')

```

```

2493     os.system(f'open {pdf_path}')
2494     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/
2495     Section_D.ly').readlines()
2496     open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/Section_D.ly', 'w'
2497     ).writelines(part_lines[15:-1])
2498
2499 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
2500     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2501     signature_copy = abjad.mutate(signatures).copy()
2502     staff_copy = abjad.mutate(staff).copy()
2503     part = abjad.Score()
2504     part.insert(0, staff)
2505     part.insert(0, signature_copy)
2506     part_file = abjad.LilyPondFile.new(
2507         part,
2508         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2509         _stylesheets/abjad.ily'],
2510         )
2511     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.')alto6'
2512     pdf_path = f'{directory}/Section_D.pdf'
2513     path = pathlib.Path('Section_D.pdf')
2514     if path.exists():
2515         print(f'Removing {pdf_path} ...')
2516         path.unlink()
2517     time_1 = time.time()
2518     print(f'Persisting {pdf_path} ...')
2519     result = abjad.persist(part_file).as_pdf(pdf_path)
2520     print(result[0])
2521     print(result[1])
2522     print(result[2])
2523     success = result[3]
2524     if success is False:
2525         print('LilyPond failed!')
2526     time_2 = time.time()
2527     total_time = time_2 - time_1
2528     print(f'Total time: {total_time} seconds')
2529     if path.exists():
2530         print(f'Opening {pdf_path} ...')
2531         os.system(f'open {pdf_path}')
2532     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.')alto6/
2533     Section_D.ly').readlines()
2534     open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.')alto6/Section_D.ly', 'w'
2535     ).writelines(part_lines[15:-1])
2536
2537 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
2538     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2539     signature_copy = abjad.mutate(signatures).copy()
2540     staff_copy = abjad.mutate(staff).copy()
2541     part = abjad.Score()
2542     part.insert(0, staff)
2543     part.insert(0, signature_copy)
2544     part_file = abjad.LilyPondFile.new(
2545         part,
2546         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2547         _stylesheets/abjad.ily'],
2548         )
2549     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.')tenor1'
2550     pdf_path = f'{directory}/Section_D.pdf'
2551     path = pathlib.Path('Section_D.pdf')

```

```

2546     if path.exists():
2547         print(f'Removing {pdf_path} ...')
2548         path.unlink()
2549     time_1 = time.time()
2550     print(f'Persisting {pdf_path} ...')
2551     result = abjad.persist(part_file).as_pdf(pdf_path)
2552     print(result[0])
2553     print(result[1])
2554     print(result[2])
2555     success = result[3]
2556     if success is False:
2557         print('LilyPond failed!')
2558     time_2 = time.time()
2559     total_time = time_2 - time_1
2560     print(f'Total time: {total_time} seconds')
2561     if path.exists():
2562         print(f'Opening {pdf_path} ...')
2563         os.system(f'open {pdf_path}')
2564     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
Section_D.ly').readlines()
2565     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Section_D.ly', ,
w').writelines(part_lines[15:-1])
2566
2567 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
2568     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2569     signature_copy = abjad.mutate(signatures).copy()
2570     staff_copy = abjad.mutate(staff).copy()
2571     part = abjad.Score()
2572     part.insert(0, staff)
2573     part.insert(0, signature_copy)
2574     part_file = abjad.LilyPondFile.new(
2575         part,
2576         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2577         )
2578     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
2579     pdf_path = f'{directory}/Section_D.pdf'
2580     path = pathlib.Path('Section_D.pdf')
2581     if path.exists():
2582         print(f'Removing {pdf_path} ...')
2583         path.unlink()
2584     time_1 = time.time()
2585     print(f'Persisting {pdf_path} ...')
2586     result = abjad.persist(part_file).as_pdf(pdf_path)
2587     print(result[0])
2588     print(result[1])
2589     print(result[2])
2590     success = result[3]
2591     if success is False:
2592         print('LilyPond failed!')
2593     time_2 = time.time()
2594     total_time = time_2 - time_1
2595     print(f'Total time: {total_time} seconds')
2596     if path.exists():
2597         print(f'Opening {pdf_path} ...')
2598         os.system(f'open {pdf_path}')
2599     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/
Section_D.ly').readlines()
2600     open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/Section_D.ly', ,

```

```

w').writelines(part_lines[15:-1])

2601 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
2602     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2603     signature_copy = abjad.mutate(signatures).copy()
2604     staff_copy = abjad.mutate(staff).copy()
2605     part = abjad.Score()
2606     part.insert(0, staff)
2607     part.insert(0, signature_copy)
2608     part_file = abjad.LilyPondFile.new(
2609         part,
2610         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2611         _stylesheets/abjad.ily'],
2612         )
2613     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3'
2614     pdf_path = f'{directory}/Section_D.pdf'
2615     path = pathlib.Path('Section_D.pdf')
2616     if path.exists():
2617         print(f'Removing {pdf_path} ...')
2618         path.unlink()
2619     time_1 = time.time()
2620     print(f'Persisting {pdf_path} ...')
2621     result = abjad.persist(part_file).as_pdf(pdf_path)
2622     print(result[0])
2623     print(result[1])
2624     print(result[2])
2625     success = result[3]
2626     if success is False:
2627         print('LilyPond failed!')
2628     time_2 = time.time()
2629     total_time = time_2 - time_1
2630     print(f'Total time: {total_time} seconds')
2631     if path.exists():
2632         print(f'Opening {pdf_path} ...')
2633         os.system(f'open {pdf_path}')
2634     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
2635     Section_D.ly').readlines()
2636     open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_D.ly', ,
2637     w').writelines(part_lines[15:-1])

2638 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
2639     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2640     signature_copy = abjad.mutate(signatures).copy()
2641     staff_copy = abjad.mutate(staff).copy()
2642     part = abjad.Score()
2643     part.insert(0, staff)
2644     part.insert(0, signature_copy)
2645     part_file = abjad.LilyPondFile.new(
2646         part,
2647         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2648         _stylesheets/abjad.ily'],
2649         )
2650     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'
2651     pdf_path = f'{directory}/Section_D.pdf'
2652     path = pathlib.Path('Section_D.pdf')
2653     if path.exists():
2654         print(f'Removing {pdf_path} ...')

```

```

2655     print(f'Persisting {pdf_path} ...')
2656     result = abjad.persist(part_file).as_pdf(pdf_path)
2657     print(result[0])
2658     print(result[1])
2659     print(result[2])
2660     success = result[3]
2661     if success is False:
2662         print('LilyPond failed!')
2663     time_2 = time.time()
2664     total_time = time_2 - time_1
2665     print(f'Total time: {total_time} seconds')
2666     if path.exists():
2667         print(f'Opening {pdf_path} ...')
2668         os.system(f'open {pdf_path}')
2669     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
2670     Section_D.ly').readlines()
2671     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_D.ly', ,
2672     w').writelines(part_lines[15:-1])
2673
2674 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
2675     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2676     signature_copy = abjad.mutate(signatures).copy()
2677     staff_copy = abjad.mutate(staff).copy()
2678     part = abjad.Score()
2679     part.insert(0, staff)
2680     part.insert(0, signature_copy)
2681     part_file = abjad.LilyPondFile.new(
2682         part,
2683         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2684         _stylesheets/abjad.ily'],
2685     )
2686     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5'
2687     pdf_path = f'{directory}/Section_D.pdf'
2688     path = pathlib.Path('Section_D.pdf')
2689     if path.exists():
2690         print(f'Removing {pdf_path} ...')
2691         path.unlink()
2692     time_1 = time.time()
2693     print(f'Persisting {pdf_path} ...')
2694     result = abjad.persist(part_file).as_pdf(pdf_path)
2695     print(result[0])
2696     print(result[1])
2697     print(result[2])
2698     success = result[3]
2699     if success is False:
2700         print('LilyPond failed!')
2701     time_2 = time.time()
2702     total_time = time_2 - time_1
2703     print(f'Total time: {total_time} seconds')
2704     if path.exists():
2705         print(f'Opening {pdf_path} ...')
2706         os.system(f'open {pdf_path}')
2707     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/
2708     Section_D.ly').readlines()
2709     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/Section_D.ly', ,
2710     w').writelines(part_lines[15:-1])
2711
2712 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
2713     signatures = abjad.select(score['Global Context']).components(abjad.Staff)

```

```

2709 signature_copy = abjad.mutate(signatures).copy()
2710 staff_copy = abjad.mutate(staff).copy()
2711 part = abjad.Score()
2712 part.insert(0, staff)
2713 part.insert(0, signature_copy)
2714 part_file = abjad.LilyPondFile.new(
2715     part,
2716     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2717 _stylesheets/abjad.ily'],
2718     )
2719 directory = '/Users/evansdsg2/Scores/guerrero/Build/part/16.)baritone1'
2720 pdf_path = f'{directory}/Section_D.pdf'
2721 path = pathlib.Path('Section_D.pdf')
2722 if path.exists():
2723     print(f'Removing {pdf_path} ...')
2724     path.unlink()
2725 time_1 = time.time()
2726 print(f'Persisting {pdf_path} ...')
2727 result = abjad.persist(part_file).as_pdf(pdf_path)
2728 print(result[0])
2729 print(result[1])
2730 print(result[2])
2731 success = result[3]
2732 if success is False:
2733     print('LilyPond failed!')
2734 time_2 = time.time()
2735 total_time = time_2 - time_1
2736 print(f'Total time: {total_time} seconds')
2737 if path.exists():
2738     print(f'Opening {pdf_path} ...')
2739     os.system(f'open {pdf_path}')
2740 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/part/16.)baritone1/
2741 Section_D.ly').readlines()
2742 open('/Users/evansdsg2/Scores/guerrero/Build/part/16.)baritone1/Section_D.ly',
2743 , 'w').writelines(part_lines[15:-1])
2744
2745 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
2746     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2747     signature_copy = abjad.mutate(signatures).copy()
2748     staff_copy = abjad.mutate(staff).copy()
2749     part = abjad.Score()
2750     part.insert(0, staff)
2751     part.insert(0, signature_copy)
2752     part_file = abjad.LilyPondFile.new(
2753         part,
2754         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2755 _stylesheets/abjad.ily'],
2756         )
2757 directory = '/Users/evansdsg2/Scores/guerrero/Build/part/17.)baritone2'
2758 pdf_path = f'{directory}/Section_D.pdf'
2759 path = pathlib.Path('Section_D.pdf')
2760 if path.exists():
2761     print(f'Removing {pdf_path} ...')
2762     path.unlink()
2763 time_1 = time.time()
2764 print(f'Persisting {pdf_path} ...')
2765 result = abjad.persist(part_file).as_pdf(pdf_path)
2766 print(result[0])
2767 print(result[1])

```

```

2764     print(result[2])
2765     success = result[3]
2766     if success is False:
2767         print('LilyPond failed!')
2768     time_2 = time.time()
2769     total_time = time_2 - time_1
2770     print(f'Total time: {total_time} seconds')
2771     if path.exists():
2772         print(f'Opening {pdf_path} ...')
2773         os.system(f'open {pdf_path}')
2774     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/
2775     Section_D.ly').readlines()
2776     open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/Section_D.ly',
2777     'w').writelines(part_lines[15:-1])
2778
2779 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
2780     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2781     signature_copy = abjad.mutate(signatures).copy()
2782     staff_copy = abjad.mutate(staff).copy()
2783     part = abjad.Score()
2784     part.insert(0, staff)
2785     part.insert(0, signature_copy)
2786     part_file = abjad.LilyPondFile.new(
2787         part,
2788         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2789         _stylesheets/abjad.ily'],
2790         )
2791     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3'
2792     pdf_path = f'{directory}/Section_D.pdf'
2793     path = pathlib.Path('Section_D.pdf')
2794     if path.exists():
2795         print(f'Removing {pdf_path} ...')
2796         path.unlink()
2797     time_1 = time.time()
2798     print(f'Persisting {pdf_path} ...')
2799     result = abjad.persist(part_file).as_pdf(pdf_path)
2800     print(result[0])
2801     print(result[1])
2802     print(result[2])
2803     success = result[3]
2804     if success is False:
2805         print('LilyPond failed!')
2806     time_2 = time.time()
2807     total_time = time_2 - time_1
2808     print(f'Total time: {total_time} seconds')
2809     if path.exists():
2810         print(f'Opening {pdf_path} ...')
2811         os.system(f'open {pdf_path}')
2812     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/
2813     Section_D.ly').readlines()
2814     open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/Section_D.ly',
2815     'w').writelines(part_lines[15:-1])
2816
2817 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
2818     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2819     signature_copy = abjad.mutate(signatures).copy()
2820     staff_copy = abjad.mutate(staff).copy()
2821     part = abjad.Score()
2822     part.insert(0, staff)

```

```

2818 part.insert(0, signature_copy)
2819 part_file = abjad.LilyPondFile.new(
2820     part,
2821     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2822 _stylesheets/abjad.ily'],
2823     )
2824 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
2825 pdf_path = f'{directory}/Section_D.pdf'
2826 path = pathlib.Path('Section_D.pdf')
2827 if path.exists():
2828     print(f'Removing {pdf_path} ...')
2829     path.unlink()
2830 time_1 = time.time()
2831 print(f'Persisting {pdf_path} ...')
2832 result = abjad.persist(part_file).as_pdf(pdf_path)
2833 print(result[0])
2834 print(result[1])
2835 print(result[2])
2836 success = result[3]
2837 if success is False:
2838     print('LilyPond failed!')
2839 time_2 = time.time()
2840 total_time = time_2 - time_1
2841 print(f'Total time: {total_time} seconds')
2842 if path.exists():
2843     print(f'Opening {pdf_path} ...')
2844     os.system(f'open {pdf_path}')
2845 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
2846 Section_D.ly').readlines()
2847 open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_D.ly', 'w'
2848 ).writelines(part_lines[15:-1])
2849
2850 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
2851     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2852     signature_copy = abjad.mutate(signatures).copy()
2853     staff_copy = abjad.mutate(staff).copy()
2854     part = abjad.Score()
2855     part.insert(0, staff)
2856     part.insert(0, signature_copy)
2857     part_file = abjad.LilyPondFile.new(
2858         part,
2859         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2860 _stylesheets/abjad.ily'],
2861         )
2862 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
2863 pdf_path = f'{directory}/Section_D.pdf'
2864 path = pathlib.Path('Section_D.pdf')
2865 if path.exists():
2866     print(f'Removing {pdf_path} ...')
2867     path.unlink()
2868 time_1 = time.time()
2869 print(f'Persisting {pdf_path} ...')
2870 result = abjad.persist(part_file).as_pdf(pdf_path)
2871 print(result[0])
2872 print(result[1])
2873 print(result[2])
2874 success = result[3]
2875 if success is False:
2876     print('LilyPond failed!')

```

```

2873 time_2 = time.time()
2874 total_time = time_2 - time_1
2875 print(f'Total time: {total_time} seconds')
2876 if path.exists():
2877     print(f'Opening {pdf_path} ...')
2878     os.system(f'open {pdf_path}')
2879 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
Section_D.ly').readlines()
2880 open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_D.ly', 'w'
').writelines(part_lines[15:-1])
2881
2882 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
2883     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2884     signature_copy = abjad.mutate(signatures).copy()
2885     staff_copy = abjad.mutate(staff).copy()
2886     part = abjad.Score()
2887     part.insert(0, staff)
2888     part.insert(0, signature_copy)
2889     part_file = abjad.LilyPondFile.new(
2890         part,
2891         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2892         )
2893     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
2894     pdf_path = f'{directory}/Section_D.pdf'
2895     path = pathlib.Path('Section_D.pdf')
2896     if path.exists():
2897         print(f'Removing {pdf_path} ...')
2898         path.unlink()
2899     time_1 = time.time()
2900     print(f'Persisting {pdf_path} ...')
2901     result = abjad.persist(part_file).as_pdf(pdf_path)
2902     print(result[0])
2903     print(result[1])
2904     print(result[2])
2905     success = result[3]
2906     if success is False:
2907         print('LilyPond failed!')
2908     time_2 = time.time()
2909     total_time = time_2 - time_1
2910     print(f'Total time: {total_time} seconds')
2911     if path.exists():
2912         print(f'Opening {pdf_path} ...')
2913         os.system(f'open {pdf_path}')
2914     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/
Section_D.ly').readlines()
2915     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_D.ly',
'w').writelines(part_lines[15:-1])

```

Listing 3.5: Invocation Source Code

## 3.6 Section E

```

1 import abjad
2 import itertools
3 import os
4 import pathlib

```

```

5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11 from TrillHandler import TrillHandler
12
13 print('Interpreting file ...')
14
15 time_signatures = [
16     abjad.TimeSignature(pair) for pair in [
17         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
18         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
19         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
20         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
21         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
22     ]
23 ]
24
25 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
26
27 def cyc(lst):
28     count = 0
29     while True:
30         yield lst[count%len(lst)]
31         count += 1
32
33 def grouper(lst1, lst2):
34     def cyc(lst):
35         c = 0
36         while True:
37             yield lst[c%len(lst)]
38             c += 1
39     lst1 = cyc(lst1)
40     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2]
41
42 def reduceMod(list_length, rw):
43     return [(x % list_length) for x in rw]
44
45 # -3 at bottom of chord for completion
46 soprano_note = [27.75, 11.5, 17.25, 8.5, 0.75, ]
47 soprano_1_note = [13.25, 16.5, 22.5, 5.25, ]
48 soprano_2_note = [13.5, 16.75, 22.25, 5.75, 16.5, ]
49 soprano_3_note = [13.25, 16.5, 22.5, 5.25, ]
50 alto_1_note = [23.75, 20.5, 12.25, 0.5, ]
51 alto_2_note = [23.5, 20.25, 12.5, 0.75, ]
52 alto_3_note = [23.25, 20.5, 12.75, 0.5, ]
53 alto_4_note = [23.5, 20.75, 12.5, 0.25, ]
54 alto_5_note = [23.75, 20.5, 12.25, 0.5, ]
55 alto_6_note = [23.5, 20.25, 12.5, 0.75, ]
56 tenor_1_note = [25.5, 6.25, 17.5, ]
57 tenor_2_note = [25.25, 6.5, 17.75, ]
58 tenor_3_note = [25.5, 6.75, 17.5, ]
59 tenor_4_note = [25.75, 6.5, 17.25, ]
60 tenor_5_note = [25.5, 6.25, 17.5, ]
61 baritone_1_note = [13.25, 24.5, 4.75, 6.5, ]
62 baritone_2_note = [13.25, 24.5, 4.75, 6.5, ]

```

```

63 baritone_3_note = [13.25, 24.5, 4.75, 6.5, ]
64 bass_1_note = [11.25, 18.5, 9.75, 0.5, ]
65 bass_2_note = [11.25, 18.5, 9.75, 0.5, ]
66 contrabass_note = [2.25, -2.5, 7.75, 18.5, 16.25, 25.5, ]
67
68 sopranino_trill = [[17, 27, ], [8, 11, ], [0, 8, ], [8, 11, ], [0, 8, ], ]
69 soprano_1_trill = [[5, 13, ], [22, 16, ], [16, 13, ]]
70 soprano_2_trill = [[22, 16, ], [5, 13, ], [16, 13, ]]
71 soprano_3_trill = [[5, 13, ], [22, 16, ], [16, 13, ]]
72 alto_1_trill = [[23, 20, ], [1, 12, ], [12, 20], [12, 23, ]]
73 alto_2_trill = [[23, 20, ], [12, 20], [1, 12, ], ]
74 alto_3_trill = [[1, 12, ], [23, 20, ], [12, 20], [12, 23, ]]
75 alto_4_trill = [[12, 20], [1, 12, ], [23, 20, ], ]
76 alto_5_trill = [[1, 12, ], [23, 20, ], [12, 23, ], [12, 20]]
77 alto_6_trill = [[23, 20, ], [1, 12, ], [12, 20]]
78 tenor_1_trill = [[-1, 6, ], [17, 25, ], [6, 17, ],]
79 tenor_2_trill = [[6, 17, ], [-1, 6, ], [17, 25, ]]
80 tenor_3_trill = [[6, 17, ], [17, 25, ], [-1, 6, ]]
81 tenor_4_trill = [[6, 17, ], [17, 25, ], [-1, 6, ]]
82 tenor_5_trill = [[-1, 6, ], [6, 17, ], [17, 25, ]]
83 baritone_1_trill = [[4, 6, ], [24, 13, ], [6, 13, ]]
84 baritone_2_trill = [[4, 6, ], [6, 13, ], [24, 13, ]]
85 baritone_3_trill = [[24, 13, ], [6, 13, ], [4, 6, ]]
86 bass_1_trill = [[0, 9, ], [18, 11, ], [11, 9, ]]
87 bass_2_trill = [[0, 9, ], [18, 11, ], [11, 9, ]]
88 contrabass_trill = [[-2, 2, ], [25, 18, ], [7, 16, ], [2, 7, ], [18, 16, ]]
89
90 def reduceMod(x, rw):
91     return [(y % x) for y in rw]
92
93 seed(1)
94 sopranino_random_walk = []
95 sopranino_random_walk.append(-1 if random() < 0.5 else 1)
96 for i in range(1, 1000):
97     movement = -1 if random() < 0.5 else 1
98     value = sopranino_random_walk[i-1] + movement
99     sopranino_random_walk.append(value)
100    sopranino_walk_chord = [11, 17, 0, 8, ]
101 l = len(sopranino_walk_chord)
102 sopranino_random_walk_notes = [sopranino_walk_chord[x] for x in reduceMod(l,
103                             sopranino_random_walk)]
104
105 seed(2)
106 soprano_1_random_walk = []
107 soprano_1_random_walk.append(-1 if random() < 0.5 else 1)
108 for i in range(1, 1000):
109     movement = -1 if random() < 0.5 else 1
110     value = soprano_1_random_walk[i-1] + movement
111     soprano_1_random_walk.append(value)
112    soprano_1_walk_chord = [13, 5, 16, 22, ]
113 l = len(soprano_1_walk_chord)
114 soprano_1_random_walk_notes = [soprano_1_walk_chord[x] for x in reduceMod(l,
115                             soprano_1_random_walk)]
116
117 seed(3)
118 soprano_2_random_walk = []
119 soprano_2_random_walk.append(-1 if random() < 0.5 else 1)

```

```

120     value = soprano_2_random_walk[i-1] + movement
121     soprano_2_random_walk.append(value)
122 soprano_2_random_walk.append(value)
123 soprano_2_walk_chord = [16, 22, 13, 5, ]
124 l = len(soprano_2_walk_chord)
125 soprano_2_random_walk_notes = [soprano_2_walk_chord[x] for x in reduceMod(l,
126                             soprano_2_random_walk)]
127
128 seed(4)
129 soprano_3_random_walk = []
130 soprano_3_random_walk.append(-1 if random() < 0.5 else 1)
131 for i in range(1, 1000):
132     movement = -1 if random() < 0.5 else 1
133     value = soprano_3_random_walk[i-1] + movement
134     soprano_3_random_walk.append(value)
135 soprano_3_random_walk.append(value)
136 soprano_3_walk_chord = [16, 5, 22, 13, ]
137 l = len(soprano_3_walk_chord)
138 soprano_3_random_walk_notes = [soprano_3_walk_chord[x] for x in reduceMod(l,
139                             soprano_3_random_walk)]
140
141 seed(5)
142 alto_1_random_walk = []
143 alto_1_random_walk.append(-1 if random() < 0.5 else 1)
144 for i in range(1, 1000):
145     movement = -1 if random() < 0.5 else 1
146     value = alto_1_random_walk[i-1] + movement
147     alto_1_random_walk.append(value)
148 alto_1_walk_chord = [12, 23, 20, 1, 12, 20, ]
149 l = len(alto_1_walk_chord)
150 alto_1_random_walk_notes = [alto_1_walk_chord[x] for x in reduceMod(l,
151                             alto_1_random_walk)]
152
153 seed(6)
154 alto_2_random_walk = []
155 alto_2_random_walk.append(-1 if random() < 0.5 else 1)
156 for i in range(1, 1000):
157     movement = -1 if random() < 0.5 else 1
158     value = alto_2_random_walk[i-1] + movement
159     alto_2_random_walk.append(value)
160 alto_2_walk_chord = [23, 20, 12, 23, 1, 12, 20, ]
161 l = len(alto_2_walk_chord)
162 alto_2_random_walk_notes = [alto_2_walk_chord[x] for x in reduceMod(l,
163                             alto_2_random_walk)]
164
165 seed(7)
166 alto_3_random_walk = []
167 alto_3_random_walk.append(-1 if random() < 0.5 else 1)
168 for i in range(1, 1000):
169     movement = -1 if random() < 0.5 else 1
170     value = alto_3_random_walk[i-1] + movement
171     alto_3_random_walk.append(value)
172 alto_3_walk_chord = [23, 20, 12, 1, 12, 20, ]
173 l = len(alto_3_walk_chord)
174 alto_3_random_walk_notes = [alto_3_walk_chord[x] for x in reduceMod(l,
175                             alto_3_random_walk)]
176
177 seed(8)
178 alto_4_random_walk = []

```

```

174 alto_4_random_walk.append(-1 if random() < 0.5 else 1)
175 for i in range(1, 1000):
176     movement = -1 if random() < 0.5 else 1
177     value = alto_4_random_walk[i-1] + movement
178     alto_4_random_walk.append(value)
179 alto_4_walk_chord = [23, 1, 12, 20, 23, 20, 12, ]
180 l = len(alto_4_walk_chord)
181 alto_4_random_walk_notes = [alto_4_walk_chord[x] for x in reduceMod(l,
182     alto_4_random_walk)]
183
184 seed(9)
185 alto_5_random_walk = []
186 alto_5_random_walk.append(-1 if random() < 0.5 else 1)
187 for i in range(1, 1000):
188     movement = -1 if random() < 0.5 else 1
189     value = alto_5_random_walk[i-1] + movement
190     alto_5_random_walk.append(value)
191 alto_5_walk_chord = [23, 1, 12, 20, 23, 20, 12, ]
192 l = len(alto_5_walk_chord)
193 alto_5_random_walk_notes = [alto_5_walk_chord[x] for x in reduceMod(l,
194     alto_5_random_walk)]
195
196 seed(10)
197 alto_6_random_walk = []
198 alto_6_random_walk.append(-1 if random() < 0.5 else 1)
199 for i in range(1, 1000):
200     movement = -1 if random() < 0.5 else 1
201     value = alto_6_random_walk[i-1] + movement
202     alto_6_random_walk.append(value)
203 alto_6_walk_chord = [23, 20, 12, 1, 12, 20, 23, ]
204 l = len(alto_6_walk_chord)
205 alto_6_random_walk_notes = [alto_6_walk_chord[x] for x in reduceMod(l,
206     alto_6_random_walk)]
207
208 seed(11)
209 tenor_1_random_walk = []
210 tenor_1_random_walk.append(-1 if random() < 0.5 else 1)
211 for i in range(1, 1000):
212     movement = -1 if random() < 0.5 else 1
213     value = tenor_1_random_walk[i-1] + movement
214     tenor_1_random_walk.append(value)
215 tenor_1_walk_chord = [-1, 17, 25, 17, 6, ]
216 l = len(tenor_1_walk_chord)
217 tenor_1_random_walk_notes = [tenor_1_walk_chord[x] for x in reduceMod(l,
218     tenor_1_random_walk)]
219
220 seed(12)
221 tenor_2_random_walk = []
222 tenor_2_random_walk.append(-1 if random() < 0.5 else 1)
223 for i in range(1, 1000):
224     movement = -1 if random() < 0.5 else 1
225     value = tenor_2_random_walk[i-1] + movement
226     tenor_2_random_walk.append(value)
227 tenor_2_walk_chord = [-1, 17, 25, 17, 6, ]
228 l = len(tenor_2_walk_chord)
229 tenor_2_random_walk_notes = [tenor_2_walk_chord[x] for x in reduceMod(l,
230     tenor_2_random_walk)]
231
232 seed(13)

```

```

228 tenor_3_random_walk = []
229 tenor_3_random_walk.append(-1 if random() < 0.5 else 1)
230 for i in range(1, 1000):
231     movement = -1 if random() < 0.5 else 1
232     value = tenor_3_random_walk[i-1] + movement
233     tenor_3_random_walk.append(value)
234 tenor_3_walk_chord = [17, 6, -1, 17, 25, ]
235 l = len(tenor_3_walk_chord)
236 tenor_3_random_walk_notes = [tenor_3_walk_chord[x] for x in reduceMod(l,
237                         tenor_3_random_walk)]
238
239 seed(14)
240 tenor_4_random_walk = []
241 tenor_4_random_walk.append(-1 if random() < 0.5 else 1)
242 for i in range(1, 1000):
243     movement = -1 if random() < 0.5 else 1
244     value = tenor_4_random_walk[i-1] + movement
245     tenor_4_random_walk.append(value)
246 tenor_4_walk_chord = [17, 6, -1, 17, 25, ]
247 l = len(tenor_4_walk_chord)
248 tenor_4_random_walk_notes = [tenor_4_walk_chord[x] for x in reduceMod(l,
249                         tenor_4_random_walk)]
250
251 seed(15)
252 tenor_5_random_walk = []
253 tenor_5_random_walk.append(-1 if random() < 0.5 else 1)
254 for i in range(1, 1000):
255     movement = -1 if random() < 0.5 else 1
256     value = tenor_5_random_walk[i-1] + movement
257     tenor_5_random_walk.append(value)
258 tenor_5_walk_chord = [25, 17, 6, -1, 17, ]
259 l = len(tenor_5_walk_chord)
260 tenor_5_random_walk_notes = [tenor_5_walk_chord[x] for x in reduceMod(l,
261                         tenor_5_random_walk)]
262
263 seed(16)
264 baritone_1_random_walk = []
265 baritone_1_random_walk.append(-1 if random() < 0.5 else 1)
266 for i in range(1, 1000):
267     movement = -1 if random() < 0.5 else 1
268     value = baritone_1_random_walk[i-1] + movement
269     baritone_1_random_walk.append(value)
270 baritone_1_walk_chord = [6, 4, 13, 24, 13, ]
271 l = len(baritone_1_walk_chord)
272 baritone_1_random_walk_notes = [baritone_1_walk_chord[x] for x in reduceMod(l,
273                         baritone_1_random_walk)]
274
275 seed(17)
276 baritone_2_random_walk = []
277 baritone_2_random_walk.append(-1 if random() < 0.5 else 1)
278 for i in range(1, 1000):
279     movement = -1 if random() < 0.5 else 1
280     value = baritone_2_random_walk[i-1] + movement
281     baritone_2_random_walk.append(value)
282 baritone_2_walk_chord = [6, 13, 4, 13, 24, ]
283 l = len(baritone_2_walk_chord)
284 baritone_2_random_walk_notes = [baritone_2_walk_chord[x] for x in reduceMod(l,
285                         baritone_2_random_walk)]

```

```

282 seed(18)
283 baritone_3_random_walk = []
284 baritone_3_random_walk.append(-1 if random() < 0.5 else 1)
285 for i in range(1, 1000):
286     movement = -1 if random() < 0.5 else 1
287     value = baritone_3_random_walk[i-1] + movement
288     baritone_3_random_walk.append(value)
289 baritone_3_walk_chord = [6, 13, 24, 13, 4, ]
290 l = len(baritone_3_walk_chord)
291 baritone_3_random_walk_notes = [baritone_3_walk_chord[x] for x in reduceMod(l,
292                             baritone_3_random_walk)]
293
294 seed(19)
295 bass_1_random_walk = []
296 bass_1_random_walk.append(-1 if random() < 0.5 else 1)
297 for i in range(1, 1000):
298     movement = -1 if random() < 0.5 else 1
299     value = bass_1_random_walk[i-1] + movement
300     bass_1_random_walk.append(value)
301 bass_1_walk_chord = [11, 9, 0, 18, ]
302 l = len(bass_1_walk_chord)
303 bass_1_random_walk_notes = [bass_1_walk_chord[x] for x in reduceMod(l,
304                             bass_1_random_walk)]
305
306 seed(20)
307 bass_2_random_walk = []
308 bass_2_random_walk.append(-1 if random() < 0.5 else 1)
309 for i in range(1, 1000):
310     movement = -1 if random() < 0.5 else 1
311     value = bass_2_random_walk[i-1] + movement
312     bass_2_random_walk.append(value)
313 bass_2_walk_chord = [0, 9, 18, 11, ]
314 l = len(bass_2_walk_chord)
315 bass_2_random_walk_notes = [bass_2_walk_chord[x] for x in reduceMod(l,
316                             bass_2_random_walk)]
317
318 seed(21)
319 contrabass_random_walk = []
320 contrabass_random_walk.append(-1 if random() < 0.5 else 1)
321 for i in range(1, 1000):
322     movement = -1 if random() < 0.5 else 1
323     value = contrabass_random_walk[i-1] + movement
324     contrabass_random_walk.append(value)
325 contrabass_walk_chord = [18, 7, 16, 2, -2, 16, 25, ]
326 l = len(contrabass_walk_chord)
327 contrabass_random_walk_notes = [contrabass_walk_chord[x] for x in reduceMod(l,
328                             contrabass_random_walk)]
329
330 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
331     talea=abjadext.rmakers.Talea(
332         counts=[12, 7, -1, 11, 8, -1, 10, 9, -1, ],
333         denominator=16,
334     ),
335     beamSpecifier=abjadext.rmakers.BeamSpecifier(
336         beamDivisionsTogether=True,
337         beamRests=False,
338     ),
339     extraCountsPerDivision=[0, 1, -1, 1, 0, -1, 0, ],
340     tupletSpecifier=abjadext.rmakers.TupletSpecifier(

```

```

337     trivialize=True,
338     extract_trivial=True,
339     rewrite_rest_filled=True,
340     rewrite_dots=True,
341     rewrite_sustained=True,
342     denominator='divisions',
343   ),
344 )
345
346 rmaker_two = abjadext.rmakers.EvenDivisionRhythmMaker(
347   denominators=[16, 16, 8, 16, 4, 8, 4, 16, 8, ],
348   extra_counts_per_division=[0, 1, -1, 0, 1, 0, -1, ],
349   logical_tie_masks=[
350     abjadext.rmakers.silence([2], 7),
351   ],
352   tupletSpecifier=abjadext.rmakers.TupletSpecifier(
353     trivialize=True,
354     extract_trivial=True,
355     rewrite_rest_filled=True,
356     rewrite_sustained=True,
357     denominator='divisions',
358   ),
359 )
360
361 attachment_handler_one = AttachmentHandler(
362   starting_dynamic='p',
363   ending_dynamic='pp',
364   hairpin='--',
365 )
366
367 attachment_handler_two = AttachmentHandler(
368   starting_dynamic='f',
369   ending_dynamic='mp',
370   hairpin='>',
371 )
372
373 attachment_handler_three = AttachmentHandler(
374   starting_dynamic='mf',
375   hairpin='--',
376 )
377
378 #####sopranino#####
379 sopranino_musicmaker_one = MusicMaker(
380   rmaker=rmaker_one,
381   pitches=sopranino_note,
382   continuous=True,
383   attachment_handler=attachment_handler_one,
384 )
385 sopranino_musicmaker_two = MusicMaker(
386   rmaker=rmaker_one,
387   pitches=sopranino_trill,
388   continuous=True,
389   attachment_handler=attachment_handler_three,
390 )
391 sopranino_musicmaker_three = MusicMaker(
392   rmaker=rmaker_two,
393   pitches=sopranino_random_walk_notes,
394   continuous=True,
395   attachment_handler=attachment_handler_two,

```

```
396 )
397 #####soprano_one#####
398 soprano_one_musicmaker_one = MusicMaker(
399     rmaker=rmaker_one,
400     pitches=soprano_1_note,
401     continuous=True,
402     attachment_handler=attachment_handler_one,
403 )
404 soprano_one_musicmaker_two = MusicMaker(
405     rmaker=rmaker_one,
406     pitches=soprano_1_trill,
407     continuous=True,
408     attachment_handler=attachment_handler_three,
409 )
410 soprano_one_musicmaker_three = MusicMaker(
411     rmaker=rmaker_two,
412     pitches=soprano_1_random_walk_notes,
413     continuous=True,
414     attachment_handler=attachment_handler_two,
415 )
416 #####soprano_two#####
417 soprano_two_musicmaker_one = MusicMaker(
418     rmaker=rmaker_one,
419     pitches=soprano_2_note,
420     continuous=True,
421     attachment_handler=attachment_handler_one,
422 )
423 soprano_two_musicmaker_two = MusicMaker(
424     rmaker=rmaker_one,
425     pitches=soprano_2_trill,
426     continuous=True,
427     attachment_handler=attachment_handler_three,
428 )
429 soprano_two_musicmaker_three = MusicMaker(
430     rmaker=rmaker_two,
431     pitches=soprano_2_random_walk_notes,
432     continuous=True,
433     attachment_handler=attachment_handler_two,
434 )
435 #####soprano_three#####
436 soprano_three_musicmaker_one = MusicMaker(
437     rmaker=rmaker_one,
438     pitches=soprano_3_note,
439     continuous=True,
440     attachment_handler=attachment_handler_one,
441 )
442 soprano_three_musicmaker_two = MusicMaker(
443     rmaker=rmaker_one,
444     pitches=soprano_3_trill,
445     continuous=True,
446     attachment_handler=attachment_handler_three,
447 )
448 soprano_three_musicmaker_three = MusicMaker(
449     rmaker=rmaker_two,
450     pitches=soprano_3_random_walk_notes,
451     continuous=True,
452     attachment_handler=attachment_handler_two,
453 )
454 #####alto_one#####
```

```
455 alto_one_musicmaker_one = MusicMaker(  
456     rmaker=rmaker_one,  
457     pitches=alto_1_note,  
458     continuous=True,  
459     attachment_handler=attachment_handler_one,  
460 )  
461 alto_one_musicmaker_two = MusicMaker(  
462     rmaker=rmaker_one,  
463     pitches=soprano_1_trill,  
464     continuous=True,  
465     attachment_handler=attachment_handler_three,  
466 )  
467 alto_one_musicmaker_three = MusicMaker(  
468     rmaker=rmaker_two,  
469     pitches=soprano_1_random_walk_notes,  
470     continuous=True,  
471     attachment_handler=attachment_handler_two,  
472 )  
473 #####alto_two#####  
474 alto_two_musicmaker_one = MusicMaker(  
475     rmaker=rmaker_one,  
476     pitches=alto_2_note,  
477     continuous=True,  
478     attachment_handler=attachment_handler_one,  
479 )  
480 alto_two_musicmaker_two = MusicMaker(  
481     rmaker=rmaker_one,  
482     pitches=soprano_2_trill,  
483     continuous=True,  
484     attachment_handler=attachment_handler_three,  
485 )  
486 alto_two_musicmaker_three = MusicMaker(  
487     rmaker=rmaker_two,  
488     pitches=soprano_2_random_walk_notes,  
489     continuous=True,  
490     attachment_handler=attachment_handler_two,  
491 )  
492 #####alto_three#####  
493 alto_three_musicmaker_one = MusicMaker(  
494     rmaker=rmaker_one,  
495     pitches=alto_3_note,  
496     continuous=True,  
497     attachment_handler=attachment_handler_one,  
498 )  
499 alto_three_musicmaker_two = MusicMaker(  
500     rmaker=rmaker_one,  
501     pitches=soprano_3_trill,  
502     continuous=True,  
503     attachment_handler=attachment_handler_three,  
504 )  
505 alto_three_musicmaker_three = MusicMaker(  
506     rmaker=rmaker_two,  
507     pitches=soprano_3_random_walk_notes,  
508     continuous=True,  
509     attachment_handler=attachment_handler_two,  
510 )  
511 #####alto_four#####  
512 alto_four_musicmaker_one = MusicMaker(  
513     rmaker=rmaker_one,
```

```
514     pitches=alto_4_note,
515     continuous=True,
516     attachment_handler=attachment_handler_one,
517 )
518 alto_four_musicmaker_two = MusicMaker(
519     rmaker=rmaker_one,
520     pitches=alto_4_trill,
521     continuous=True,
522     attachment_handler=attachment_handler_three,
523 )
524 alto_four_musicmaker_three = MusicMaker(
525     rmaker=rmaker_two,
526     pitches=alto_4_random_walk_notes,
527     continuous=True,
528     attachment_handler=attachment_handler_two,
529 )
530 #####alto_five#####
531 alto_five_musicmaker_one = MusicMaker(
532     rmaker=rmaker_one,
533     pitches=alto_5_note,
534     continuous=True,
535     attachment_handler=attachment_handler_one,
536 )
537 alto_five_musicmaker_two = MusicMaker(
538     rmaker=rmaker_one,
539     pitches=alto_5_trill,
540     continuous=True,
541     attachment_handler=attachment_handler_three,
542 )
543 alto_five_musicmaker_three = MusicMaker(
544     rmaker=rmaker_two,
545     pitches=alto_5_random_walk_notes,
546     continuous=True,
547     attachment_handler=attachment_handler_two,
548 )
549 #####alto_six#####
550 alto_six_musicmaker_one = MusicMaker(
551     rmaker=rmaker_one,
552     pitches=alto_6_note,
553     continuous=True,
554     attachment_handler=attachment_handler_one,
555 )
556 alto_six_musicmaker_two = MusicMaker(
557     rmaker=rmaker_one,
558     pitches=alto_6_trill,
559     continuous=True,
560     attachment_handler=attachment_handler_three,
561 )
562 alto_six_musicmaker_three = MusicMaker(
563     rmaker=rmaker_two,
564     pitches=alto_6_random_walk_notes,
565     continuous=True,
566     attachment_handler=attachment_handler_two,
567 )
568 #####tenor_one#####
569 tenor_one_musicmaker_one = MusicMaker(
570     rmaker=rmaker_one,
571     pitches=tenor_1_note,
572     continuous=True,
```

```
573     attachment_handler=attachment_handler_one,
574 )
575 tenor_one_musicmaker_two = MusicMaker(
576     rmaker=rmaker_one,
577     pitches=tenor_1_trill,
578     continuous=True,
579     attachment_handler=attachment_handler_three,
580 )
581 tenor_one_musicmaker_three = MusicMaker(
582     rmaker=rmaker_two,
583     pitches=tenor_1_random_walk_notes,
584     continuous=True,
585     attachment_handler=attachment_handler_two,
586 )
587 #####tenor_two#####
588 tenor_two_musicmaker_one = MusicMaker(
589     rmaker=rmaker_one,
590     pitches=tenor_2_note,
591     continuous=True,
592     attachment_handler=attachment_handler_one,
593 )
594 tenor_two_musicmaker_two = MusicMaker(
595     rmaker=rmaker_one,
596     pitches=tenor_2_trill,
597     continuous=True,
598     attachment_handler=attachment_handler_three,
599 )
600 tenor_two_musicmaker_three = MusicMaker(
601     rmaker=rmaker_two,
602     pitches=tenor_2_random_walk_notes,
603     continuous=True,
604     attachment_handler=attachment_handler_two,
605 )
606 #####tenor_three#####
607 tenor_three_musicmaker_one = MusicMaker(
608     rmaker=rmaker_one,
609     pitches=tenor_3_note,
610     continuous=True,
611     attachment_handler=attachment_handler_one,
612 )
613 tenor_three_musicmaker_two = MusicMaker(
614     rmaker=rmaker_one,
615     pitches=tenor_3_trill,
616     continuous=True,
617     attachment_handler=attachment_handler_three,
618 )
619 tenor_three_musicmaker_three = MusicMaker(
620     rmaker=rmaker_two,
621     pitches=tenor_3_random_walk_notes,
622     continuous=True,
623     attachment_handler=attachment_handler_two,
624 )
625 #####tenor_four#####
626 tenor_four_musicmaker_one = MusicMaker(
627     rmaker=rmaker_one,
628     pitches=tenor_4_note,
629     continuous=True,
630     attachment_handler=attachment_handler_one,
631 )
```

```
632 tenor_four_musicmaker_two = MusicMaker(  
633     rmaker=rmaker_one,  
634     pitches=tenor_4_trill,  
635     continuous=True,  
636     attachment_handler=attachment_handler_three,  
637 )  
638 tenor_four_musicmaker_three = MusicMaker(  
639     rmaker=rmaker_two,  
640     pitches=tenor_4_random_walk_notes,  
641     continuous=True,  
642     attachment_handler=attachment_handler_two,  
643 )  
644 #####tenor_five#####  
645 tenor_five_musicmaker_one = MusicMaker(  
646     rmaker=rmaker_one,  
647     pitches=tenor_5_note,  
648     continuous=True,  
649     attachment_handler=attachment_handler_one,  
650 )  
651 tenor_five_musicmaker_two = MusicMaker(  
652     rmaker=rmaker_one,  
653     pitches=tenor_5_trill,  
654     continuous=True,  
655     attachment_handler=attachment_handler_three,  
656 )  
657 tenor_five_musicmaker_three = MusicMaker(  
658     rmaker=rmaker_two,  
659     pitches=tenor_5_random_walk_notes,  
660     continuous=True,  
661     attachment_handler=attachment_handler_two,  
662 )  
663 #####baritone_one#####  
664 baritone_one_musicmaker_one = MusicMaker(  
665     rmaker=rmaker_one,  
666     pitches=baritone_1_note,  
667     continuous=True,  
668     attachment_handler=attachment_handler_one,  
669 )  
670 baritone_one_musicmaker_two = MusicMaker(  
671     rmaker=rmaker_one,  
672     pitches=baritone_1_trill,  
673     continuous=True,  
674     attachment_handler=attachment_handler_three,  
675 )  
676 baritone_one_musicmaker_three = MusicMaker(  
677     rmaker=rmaker_two,  
678     pitches=baritone_1_random_walk_notes,  
679     continuous=True,  
680     attachment_handler=attachment_handler_two,  
681 )  
682 #####baritone_two#####  
683 baritone_two_musicmaker_one = MusicMaker(  
684     rmaker=rmaker_one,  
685     pitches=baritone_2_note,  
686     continuous=True,  
687     attachment_handler=attachment_handler_one,  
688 )  
689 baritone_two_musicmaker_two = MusicMaker(  
690     rmaker=rmaker_one,
```

```
691     pitches=baritone_2_trill,
692     continuous=True,
693     attachment_handler=attachment_handler_three,
694 )
695 baritone_two_musicmaker_three = MusicMaker(
696     rmaker=rmaker_two,
697     pitches=baritone_2_random_walk_notes,
698     continuous=True,
699     attachment_handler=attachment_handler_two,
700 )
701 #####baritone_three#####
702 baritone_three_musicmaker_one = MusicMaker(
703     rmaker=rmaker_one,
704     pitches=baritone_3_note,
705     continuous=True,
706     attachment_handler=attachment_handler_one,
707 )
708 baritone_three_musicmaker_two = MusicMaker(
709     rmaker=rmaker_one,
710     pitches=baritone_3_trill,
711     continuous=True,
712     attachment_handler=attachment_handler_three,
713 )
714 baritone_three_musicmaker_three = MusicMaker(
715     rmaker=rmaker_two,
716     pitches=baritone_3_random_walk_notes,
717     continuous=True,
718     attachment_handler=attachment_handler_two,
719 )
720 #####bass_one#####
721 bass_one_musicmaker_one = MusicMaker(
722     rmaker=rmaker_one,
723     pitches=bass_1_note,
724     continuous=True,
725     attachment_handler=attachment_handler_one,
726 )
727 bass_one_musicmaker_two = MusicMaker(
728     rmaker=rmaker_one,
729     pitches=bass_1_trill,
730     continuous=True,
731     attachment_handler=attachment_handler_three,
732 )
733 bass_one_musicmaker_three = MusicMaker(
734     rmaker=rmaker_two,
735     pitches=bass_1_random_walk_notes,
736     continuous=True,
737     attachment_handler=attachment_handler_two,
738 )
739 #####bass_two#####
740 bass_two_musicmaker_one = MusicMaker(
741     rmaker=rmaker_one,
742     pitches=bass_2_note,
743     continuous=True,
744     attachment_handler=attachment_handler_one,
745 )
746 bass_two_musicmaker_two = MusicMaker(
747     rmaker=rmaker_one,
748     pitches=bass_2_trill,
749     continuous=True,
```

```

750     attachment_handler=attachment_handler_three,
751 )
752 bass_two_musicmaker_three = MusicMaker(
753     rmaker=rmaker_two,
754     pitches=bass_2_random_walk_notes,
755     continuous=True,
756     attachment_handler=attachment_handler_two,
757 )
758 #####contrabass#####
759 contrabass_musicmaker_one = MusicMaker(
760     rmaker=rmaker_one,
761     pitches=contrabass_note,
762     continuous=True,
763     attachment_handler=attachment_handler_one,
764 )
765 contrabass_musicmaker_two = MusicMaker(
766     rmaker=rmaker_one,
767     pitches=contrabass_trill,
768     continuous=True,
769     attachment_handler=attachment_handler_three,
770 )
771 contrabass_musicmaker_three = MusicMaker(
772     rmaker=rmaker_two,
773     pitches=contrabass_random_walk_notes,
774     continuous=True,
775     attachment_handler=attachment_handler_two,
776 )
777
778 silence_maker = abjadext.rmakers.NoteRhythmMaker(
779     division_masks=[
780         abjadext.rmakers.SilenceMask(
781             pattern=abjad.index([0], 1),
782             ),
783         ],
784     )
785
786 class MusicSpecifier:
787
788     def __init__(self, music_maker, voice_name):
789         self.music_maker = music_maker
790         self.voice_name = voice_name
791
792 print('Collecting timespans and rmakers ...')
793
794 voice_1_timespan_list = abjad.TimespanList([
795     abjad.AnnotatedTimespan(
796         start_offset=start_offset,
797         stop_offset=stop_offset,
798         annotation=MusicSpecifier(
799             music_maker=music_maker,
800             voice_name='Voice 1',
801             ),
802     )
803     for start_offset, stop_offset, music_maker in [
804         [(0, 8), (2, 8), soprano_musicmaker_three],
805         [(2, 8), (4, 8), soprano_musicmaker_three],
806         [(4, 8), (6, 8), soprano_musicmaker_three],
807         [(6, 8), (8, 8), soprano_musicmaker_three],
808         [(8, 8), (10, 8), soprano_musicmaker_two],

```

```
809      [(10, 8), (12, 8), sopranino_musicmaker_two],  
810      [(12, 8), (14, 8), sopranino_musicmaker_two],  
811      [(14, 8), (16, 8), sopranino_musicmaker_three],  
812      [(16, 8), (18, 8), sopranino_musicmaker_three],  
813      [(18, 8), (20, 8), sopranino_musicmaker_one],  
814      [(20, 8), (22, 8), sopranino_musicmaker_one],  
815      [(22, 8), (24, 8), sopranino_musicmaker_one],  
816      [(24, 8), (26, 8), sopranino_musicmaker_three],  
817      [(26, 8), (28, 8), sopranino_musicmaker_three],  
818      [(28, 8), (30, 8), sopranino_musicmaker_three],  
819      [(30, 8), (32, 8), sopranino_musicmaker_three],  
820      [(32, 8), (34, 8), sopranino_musicmaker_three],  
821      [(34, 8), (36, 8), sopranino_musicmaker_one],  
822      [(36, 8), (38, 8), sopranino_musicmaker_two],  
823      [(38, 8), (40, 8), sopranino_musicmaker_one],  
824      [(40, 8), (42, 8), sopranino_musicmaker_two],  
825      [(42, 8), (44, 8), sopranino_musicmaker_one],  
826      [(44, 8), (46, 8), sopranino_musicmaker_one],  
827      [(46, 8), (48, 8), sopranino_musicmaker_one],  
828      [(48, 8), (50, 8), sopranino_musicmaker_two],  
829      [(50, 8), (52, 8), sopranino_musicmaker_two],  
830      [(52, 8), (54, 8), sopranino_musicmaker_two],  
831      [(54, 8), (56, 8), sopranino_musicmaker_three],  
832      [(56, 8), (58, 8), sopranino_musicmaker_three],  
833      [(58, 8), (60, 8), sopranino_musicmaker_one],  
834      [(60, 8), (62, 8), sopranino_musicmaker_three],  
835      [(62, 8), (64, 8), sopranino_musicmaker_two],  
836      [(64, 8), (66, 8), sopranino_musicmaker_two],  
837      [(66, 8), (68, 8), sopranino_musicmaker_three],  
838      [(68, 8), (70, 8), sopranino_musicmaker_one],  
839      [(70, 8), (72, 8), sopranino_musicmaker_one],  
840      [(72, 8), (74, 8), sopranino_musicmaker_three],  
841      [(74, 8), (76, 8), sopranino_musicmaker_three],  
842      [(76, 8), (78, 8), sopranino_musicmaker_two],  
843      [(78, 8), (80, 8), sopranino_musicmaker_two],  
844      [(80, 8), (82, 8), sopranino_musicmaker_one],  
845      [(82, 8), (84, 8), sopranino_musicmaker_three],  
846      [(84, 8), (86, 8), sopranino_musicmaker_two],  
847      [(86, 8), (88, 8), sopranino_musicmaker_two],  
848      [(88, 8), (90, 8), sopranino_musicmaker_three],  
849      [(90, 8), (92, 8), sopranino_musicmaker_one],  
850      [(92, 8), (94, 8), sopranino_musicmaker_three],  
851      [(94, 8), (96, 8), sopranino_musicmaker_three],  
852      [(96, 8), (98, 8), sopranino_musicmaker_two],  
853      [(98, 8), (100, 8), sopranino_musicmaker_two],  
854      [(100, 8), (102, 8), sopranino_musicmaker_one],  
855      [(102, 8), (104, 8), sopranino_musicmaker_three],  
856      [(104, 8), (106, 8), sopranino_musicmaker_three],  
857      [(106, 8), (108, 8), sopranino_musicmaker_three],  
858      [(108, 8), (110, 8), sopranino_musicmaker_two],  
859      [(110, 8), (112, 8), sopranino_musicmaker_two],  
860      [(112, 8), (114, 8), sopranino_musicmaker_two],  
861      [(114, 8), (116, 8), sopranino_musicmaker_one],  
862      [(116, 8), (118, 8), sopranino_musicmaker_three],  
863      [(118, 8), (120, 8), sopranino_musicmaker_three],  
864      [(120, 8), (122, 8), sopranino_musicmaker_three],  
865      [(122, 8), (124, 8), sopranino_musicmaker_two],  
866      [(124, 8), (126, 8), sopranino_musicmaker_three],  
867      [(126, 8), (128, 8), sopranino_musicmaker_one],
```

```

868     [(128, 8), (130, 8), sopranino_musicmaker_one],
869     [(130, 8), (132, 8), sopranino_musicmaker_one],
870     [(132, 8), (134, 8), sopranino_musicmaker_three],
871     [(134, 8), (136, 8), sopranino_musicmaker_three],
872     [(136, 8), (138, 8), sopranino_musicmaker_three],
873     [(138, 8), (140, 8), sopranino_musicmaker_three],
874     [(140, 8), (142, 8), sopranino_musicmaker_three],
875     [(142, 8), (144, 8), sopranino_musicmaker_three],
876     [(144, 8), (146, 8), sopranino_musicmaker_three],
877     [(146, 8), (148, 8), sopranino_musicmaker_three],
878     [(148, 8), (150, 8), sopranino_musicmaker_one],
879     [(150, 8), (152, 8), sopranino_musicmaker_one],
880     [(152, 8), (154, 8), sopranino_musicmaker_two],
881     [(154, 8), (156, 8), sopranino_musicmaker_three],
882     [(156, 8), (158, 8), sopranino_musicmaker_two],
883     [(158, 8), (160, 8), sopranino_musicmaker_two],
884     [(160, 8), (162, 8), sopranino_musicmaker_three],
885     [(162, 8), (164, 8), sopranino_musicmaker_three],
886     [(164, 8), (166, 8), sopranino_musicmaker_one],
887     [(166, 8), (168, 8), sopranino_musicmaker_three],
888     [(168, 8), (170, 8), sopranino_musicmaker_three],
889     [(170, 8), (172, 8), sopranino_musicmaker_three],
890     [(172, 8), (174, 8), sopranino_musicmaker_three],
891     [(174, 8), (176, 8), sopranino_musicmaker_three],
892     [(176, 8), (178, 8), sopranino_musicmaker_three],
893     [(178, 8), (180, 8), sopranino_musicmaker_three],
894     [(180, 8), (182, 8), sopranino_musicmaker_three],
895     [(182, 8), (184, 8), sopranino_musicmaker_three],
896     [(184, 8), (186, 8), sopranino_musicmaker_three],
897     [(186, 8), (188, 8), sopranino_musicmaker_one],
898     [(188, 8), (190, 8), sopranino_musicmaker_three],
899     [(190, 8), (192, 8), sopranino_musicmaker_two],
900     [(192, 8), (194, 8), sopranino_musicmaker_two],
901     [(194, 8), (196, 8), sopranino_musicmaker_one],
902     [(196, 8), (198, 8), sopranino_musicmaker_one],
903     [(198, 8), (199, 8), sopranino_musicmaker_one],
904     [(199, 8), (200, 8), sopranino_musicmaker_one],
905 ]
906 ])
907
908 voice_2_timespan_list = abjad.TimespanList([
909     abjad.AnnotatedTimespan(
910         start_offset=start_offset,
911         stop_offset=stop_offset,
912         annotation=MusicSpecifier(
913             music_maker=music_maker,
914             voice_name='Voice 2',
915         ),
916     )
917     for start_offset, stop_offset, music_maker in [
918         [(0, 8), (2, 8), soprano_one_musicmaker_three],
919         [(2, 8), (4, 8), soprano_one_musicmaker_three],
920         [(4, 8), (6, 8), soprano_one_musicmaker_one],
921         [(6, 8), (8, 8), soprano_one_musicmaker_three],
922         [(8, 8), (10, 8), soprano_one_musicmaker_one],
923         [(10, 8), (12, 8), soprano_one_musicmaker_one],
924         [(12, 8), (14, 8), soprano_one_musicmaker_three],
925         [(14, 8), (16, 8), soprano_one_musicmaker_three],
926         [(16, 8), (18, 8), soprano_one_musicmaker_two],

```

```
927 [(18, 8), (20, 8), soprano_one_musicmaker_two],  
928 [(20, 8), (22, 8), soprano_one_musicmaker_three],  
929 [(22, 8), (24, 8), soprano_one_musicmaker_three],  
930 [(24, 8), (26, 8), soprano_one_musicmaker_one],  
931 [(26, 8), (28, 8), soprano_one_musicmaker_three],  
932 [(28, 8), (30, 8), soprano_one_musicmaker_two],  
933 [(30, 8), (32, 8), soprano_one_musicmaker_one],  
934 [(32, 8), (34, 8), soprano_one_musicmaker_three],  
935 [(34, 8), (36, 8), soprano_one_musicmaker_three],  
936 [(36, 8), (38, 8), soprano_one_musicmaker_three],  
937 [(38, 8), (40, 8), soprano_one_musicmaker_three],  
938 [(40, 8), (42, 8), soprano_one_musicmaker_one],  
939 [(42, 8), (44, 8), soprano_one_musicmaker_three],  
940 [(44, 8), (46, 8), soprano_one_musicmaker_two],  
941 [(46, 8), (48, 8), soprano_one_musicmaker_two],  
942 [(48, 8), (50, 8), soprano_one_musicmaker_two],  
943 [(50, 8), (52, 8), soprano_one_musicmaker_three],  
944 [(52, 8), (54, 8), soprano_one_musicmaker_one],  
945 [(54, 8), (56, 8), soprano_one_musicmaker_one],  
946 [(56, 8), (58, 8), soprano_one_musicmaker_three],  
947 [(58, 8), (60, 8), soprano_one_musicmaker_three],  
948 [(60, 8), (62, 8), soprano_one_musicmaker_three],  
949 [(62, 8), (64, 8), soprano_one_musicmaker_two],  
950 [(64, 8), (66, 8), soprano_one_musicmaker_two],  
951 [(66, 8), (68, 8), soprano_one_musicmaker_one],  
952 [(68, 8), (70, 8), soprano_one_musicmaker_one],  
953 [(70, 8), (72, 8), soprano_one_musicmaker_one],  
954 [(72, 8), (74, 8), soprano_one_musicmaker_three],  
955 [(74, 8), (76, 8), soprano_one_musicmaker_three],  
956 [(76, 8), (78, 8), soprano_one_musicmaker_three],  
957 [(78, 8), (80, 8), soprano_one_musicmaker_two],  
958 [(80, 8), (82, 8), soprano_one_musicmaker_two],  
959 [(82, 8), (84, 8), soprano_one_musicmaker_two],  
960 [(84, 8), (86, 8), soprano_one_musicmaker_three],  
961 [(86, 8), (88, 8), soprano_one_musicmaker_three],  
962 [(88, 8), (90, 8), soprano_one_musicmaker_three],  
963 [(90, 8), (92, 8), soprano_one_musicmaker_one],  
964 [(92, 8), (94, 8), soprano_one_musicmaker_three],  
965 [(94, 8), (96, 8), soprano_one_musicmaker_three],  
966 [(96, 8), (98, 8), soprano_one_musicmaker_one],  
967 [(98, 8), (100, 8), soprano_one_musicmaker_two],  
968 [(100, 8), (102, 8), soprano_one_musicmaker_two],  
969 [(102, 8), (104, 8), soprano_one_musicmaker_three],  
970 [(104, 8), (106, 8), soprano_one_musicmaker_three],  
971 [(106, 8), (108, 8), soprano_one_musicmaker_three],  
972 [(108, 8), (110, 8), soprano_one_musicmaker_three],  
973 [(110, 8), (112, 8), soprano_one_musicmaker_three],  
974 [(112, 8), (114, 8), soprano_one_musicmaker_one],  
975 [(114, 8), (116, 8), soprano_one_musicmaker_three],  
976 [(116, 8), (118, 8), soprano_one_musicmaker_two],  
977 [(118, 8), (120, 8), soprano_one_musicmaker_two],  
978 [(120, 8), (122, 8), soprano_one_musicmaker_three],  
979 [(122, 8), (124, 8), soprano_one_musicmaker_three],  
980 [(124, 8), (126, 8), soprano_one_musicmaker_two],  
981 [(126, 8), (128, 8), soprano_one_musicmaker_three],  
982 [(128, 8), (130, 8), soprano_one_musicmaker_one],  
983 [(130, 8), (132, 8), soprano_one_musicmaker_one],  
984 [(132, 8), (134, 8), soprano_one_musicmaker_two],  
985 [(134, 8), (136, 8), soprano_one_musicmaker_two],
```

```

986     [(136, 8), (138, 8), soprano_one_musicmaker_three],
987     [(138, 8), (140, 8), soprano_one_musicmaker_three],
988     [(140, 8), (142, 8), soprano_one_musicmaker_three],
989     [(142, 8), (144, 8), soprano_one_musicmaker_three],
990     [(144, 8), (146, 8), soprano_one_musicmaker_three],
991     [(146, 8), (148, 8), soprano_one_musicmaker_one],
992     [(148, 8), (150, 8), soprano_one_musicmaker_three],
993     [(150, 8), (152, 8), soprano_one_musicmaker_one],
994     [(152, 8), (154, 8), soprano_one_musicmaker_one],
995     [(154, 8), (156, 8), soprano_one_musicmaker_one],
996     [(156, 8), (158, 8), soprano_one_musicmaker_three],
997     [(158, 8), (160, 8), soprano_one_musicmaker_three],
998     [(160, 8), (162, 8), soprano_one_musicmaker_three],
999     [(162, 8), (164, 8), soprano_one_musicmaker_one],
1000    [(164, 8), (166, 8), soprano_one_musicmaker_one],
1001    [(166, 8), (168, 8), soprano_one_musicmaker_three],
1002    [(168, 8), (170, 8), soprano_one_musicmaker_three],
1003    [(170, 8), (172, 8), soprano_one_musicmaker_three],
1004    [(172, 8), (174, 8), soprano_one_musicmaker_three],
1005    [(174, 8), (176, 8), soprano_one_musicmaker_one],
1006    [(176, 8), (178, 8), soprano_one_musicmaker_one],
1007    [(178, 8), (180, 8), soprano_one_musicmaker_three],
1008    [(180, 8), (182, 8), soprano_one_musicmaker_one],
1009    [(182, 8), (184, 8), soprano_one_musicmaker_three],
1010    [(184, 8), (186, 8), soprano_one_musicmaker_three],
1011    [(186, 8), (188, 8), soprano_one_musicmaker_three],
1012    [(188, 8), (190, 8), soprano_one_musicmaker_two],
1013    [(190, 8), (192, 8), soprano_one_musicmaker_two],
1014    [(192, 8), (194, 8), soprano_one_musicmaker_two],
1015    [(194, 8), (196, 8), soprano_one_musicmaker_two],
1016    [(196, 8), (198, 8), soprano_one_musicmaker_two],
1017    [(198, 8), (199, 8), soprano_one_musicmaker_one],
1018    [(199, 8), (200, 8), soprano_one_musicmaker_three],
1019 ]
1020 ])
1021
1022 voice_3_timestspan_list = abjad.TimespanList([
1023     abjad.AnnotatedTimespan(
1024         start_offset=start_offset,
1025         stop_offset=stop_offset,
1026         annotation=MusicSpecifier(
1027             music_maker=music_maker,
1028             voice_name='Voice 3',
1029         ),
1030     )
1031     for start_offset, stop_offset, music_maker in [
1032         [(0, 8), (2, 8), soprano_two_musicmaker_three],
1033         [(2, 8), (4, 8), soprano_two_musicmaker_three],
1034         [(4, 8), (6, 8), soprano_two_musicmaker_three],
1035         [(6, 8), (8, 8), soprano_two_musicmaker_two],
1036         [(8, 8), (10, 8), soprano_two_musicmaker_two],
1037         [(10, 8), (12, 8), soprano_two_musicmaker_two],
1038         [(12, 8), (14, 8), soprano_two_musicmaker_two],
1039         [(14, 8), (16, 8), soprano_two_musicmaker_three],
1040         [(16, 8), (18, 8), soprano_two_musicmaker_three],
1041         [(18, 8), (20, 8), soprano_two_musicmaker_three],
1042         [(20, 8), (22, 8), soprano_two_musicmaker_three],
1043         [(22, 8), (24, 8), soprano_two_musicmaker_one],
1044         [(24, 8), (26, 8), soprano_two_musicmaker_one],

```

```
1045 [(26, 8), (28, 8), soprano_two_musicmaker_three],  
1046 [(28, 8), (30, 8), soprano_two_musicmaker_three],  
1047 [(30, 8), (32, 8), soprano_two_musicmaker_three],  
1048 [(32, 8), (34, 8), soprano_two_musicmaker_two],  
1049 [(34, 8), (36, 8), soprano_two_musicmaker_two],  
1050 [(36, 8), (38, 8), soprano_two_musicmaker_two],  
1051 [(38, 8), (40, 8), soprano_two_musicmaker_one],  
1052 [(40, 8), (42, 8), soprano_two_musicmaker_one],  
1053 [(42, 8), (44, 8), soprano_two_musicmaker_one],  
1054 [(44, 8), (46, 8), soprano_two_musicmaker_three],  
1055 [(46, 8), (48, 8), soprano_two_musicmaker_three],  
1056 [(48, 8), (50, 8), soprano_two_musicmaker_one],  
1057 [(50, 8), (52, 8), soprano_two_musicmaker_three],  
1058 [(52, 8), (54, 8), soprano_two_musicmaker_three],  
1059 [(54, 8), (56, 8), soprano_two_musicmaker_three],  
1060 [(56, 8), (58, 8), soprano_two_musicmaker_three],  
1061 [(58, 8), (60, 8), soprano_two_musicmaker_three],  
1062 [(60, 8), (62, 8), soprano_two_musicmaker_three],  
1063 [(62, 8), (64, 8), soprano_two_musicmaker_three],  
1064 [(64, 8), (66, 8), soprano_two_musicmaker_two],  
1065 [(66, 8), (68, 8), soprano_two_musicmaker_two],  
1066 [(68, 8), (70, 8), soprano_two_musicmaker_two],  
1067 [(70, 8), (72, 8), soprano_two_musicmaker_two],  
1068 [(72, 8), (74, 8), soprano_two_musicmaker_two],  
1069 [(74, 8), (76, 8), soprano_two_musicmaker_three],  
1070 [(76, 8), (78, 8), soprano_two_musicmaker_three],  
1071 [(78, 8), (80, 8), soprano_two_musicmaker_three],  
1072 [(80, 8), (82, 8), soprano_two_musicmaker_three],  
1073 [(82, 8), (84, 8), soprano_two_musicmaker_three],  
1074 [(84, 8), (86, 8), soprano_two_musicmaker_three],  
1075 [(86, 8), (88, 8), soprano_two_musicmaker_one],  
1076 [(88, 8), (90, 8), soprano_two_musicmaker_one],  
1077 [(90, 8), (92, 8), soprano_two_musicmaker_three],  
1078 [(92, 8), (94, 8), soprano_two_musicmaker_two],  
1079 [(94, 8), (96, 8), soprano_two_musicmaker_three],  
1080 [(96, 8), (98, 8), soprano_two_musicmaker_three],  
1081 [(98, 8), (100, 8), soprano_two_musicmaker_three],  
1082 [(100, 8), (102, 8), soprano_two_musicmaker_one],  
1083 [(102, 8), (104, 8), soprano_two_musicmaker_three],  
1084 [(104, 8), (106, 8), soprano_two_musicmaker_three],  
1085 [(106, 8), (108, 8), soprano_two_musicmaker_three],  
1086 [(108, 8), (110, 8), soprano_two_musicmaker_three],  
1087 [(110, 8), (112, 8), soprano_two_musicmaker_two],  
1088 [(112, 8), (114, 8), soprano_two_musicmaker_two],  
1089 [(114, 8), (116, 8), soprano_two_musicmaker_three],  
1090 [(116, 8), (118, 8), soprano_two_musicmaker_three],  
1091 [(118, 8), (120, 8), soprano_two_musicmaker_one],  
1092 [(120, 8), (122, 8), soprano_two_musicmaker_one],  
1093 [(122, 8), (124, 8), soprano_two_musicmaker_three],  
1094 [(124, 8), (126, 8), soprano_two_musicmaker_three],  
1095 [(126, 8), (128, 8), soprano_two_musicmaker_one],  
1096 [(128, 8), (130, 8), soprano_two_musicmaker_three],  
1097 [(130, 8), (132, 8), soprano_two_musicmaker_three],  
1098 [(132, 8), (134, 8), soprano_two_musicmaker_one],  
1099 [(134, 8), (136, 8), soprano_two_musicmaker_one],  
1100 [(136, 8), (138, 8), soprano_two_musicmaker_three],  
1101 [(138, 8), (140, 8), soprano_two_musicmaker_three],  
1102 [(140, 8), (142, 8), soprano_two_musicmaker_three],  
1103 [(142, 8), (144, 8), soprano_two_musicmaker_three],
```

```

1104     [(144, 8), (146, 8), soprano_two_musicmaker_one],
1105     [(146, 8), (148, 8), soprano_two_musicmaker_one],
1106     [(148, 8), (150, 8), soprano_two_musicmaker_two],
1107     [(150, 8), (152, 8), soprano_two_musicmaker_two],
1108     [(152, 8), (154, 8), soprano_two_musicmaker_two],
1109     [(154, 8), (156, 8), soprano_two_musicmaker_three],
1110     [(156, 8), (158, 8), soprano_two_musicmaker_three],
1111     [(158, 8), (160, 8), soprano_two_musicmaker_three],
1112     [(160, 8), (162, 8), soprano_two_musicmaker_three],
1113     [(162, 8), (164, 8), soprano_two_musicmaker_three],
1114     [(164, 8), (166, 8), soprano_two_musicmaker_three],
1115     [(166, 8), (168, 8), soprano_two_musicmaker_three],
1116     [(168, 8), (170, 8), soprano_two_musicmaker_one],
1117     [(170, 8), (172, 8), soprano_two_musicmaker_three],
1118     [(172, 8), (174, 8), soprano_two_musicmaker_one],
1119     [(174, 8), (176, 8), soprano_two_musicmaker_one],
1120     [(176, 8), (178, 8), soprano_two_musicmaker_three],
1121     [(178, 8), (180, 8), soprano_two_musicmaker_three],
1122     [(180, 8), (182, 8), soprano_two_musicmaker_two],
1123     [(182, 8), (184, 8), soprano_two_musicmaker_two],
1124     [(184, 8), (186, 8), soprano_two_musicmaker_two],
1125     [(186, 8), (188, 8), soprano_two_musicmaker_three],
1126     [(188, 8), (190, 8), soprano_two_musicmaker_one],
1127     [(190, 8), (192, 8), soprano_two_musicmaker_three],
1128     [(192, 8), (194, 8), soprano_two_musicmaker_three],
1129     [(194, 8), (196, 8), soprano_two_musicmaker_three],
1130     [(196, 8), (198, 8), soprano_two_musicmaker_three],
1131     [(198, 8), (199, 8), soprano_two_musicmaker_three],
1132     [(199, 8), (200, 8), soprano_two_musicmaker_three],
1133 ]
1134 ])
1135
1136 voice_4_timespan_list = abjad.TimespanList([
1137     abjad.AnnotatedTimespan(
1138         start_offset=start_offset,
1139         stop_offset=stop_offset,
1140         annotation=MusicSpecifier(
1141             music_maker=music_maker,
1142             voice_name='Voice 4',
1143         ),
1144     )
1145     for start_offset, stop_offset, music_maker in [
1146         [(0, 8), (2, 8), soprano_three_musicmaker_three],
1147         [(2, 8), (4, 8), soprano_three_musicmaker_three],
1148         [(4, 8), (6, 8), soprano_three_musicmaker_one],
1149         [(6, 8), (8, 8), soprano_three_musicmaker_one],
1150         [(8, 8), (10, 8), soprano_three_musicmaker_one],
1151         [(10, 8), (12, 8), soprano_three_musicmaker_three],
1152         [(12, 8), (14, 8), soprano_three_musicmaker_three],
1153         [(14, 8), (16, 8), soprano_three_musicmaker_three],
1154         [(16, 8), (18, 8), soprano_three_musicmaker_three],
1155         [(18, 8), (20, 8), soprano_three_musicmaker_three],
1156         [(20, 8), (22, 8), soprano_three_musicmaker_one],
1157         [(22, 8), (24, 8), soprano_three_musicmaker_one],
1158         [(24, 8), (26, 8), soprano_three_musicmaker_three],
1159         [(26, 8), (28, 8), soprano_three_musicmaker_three],
1160         [(28, 8), (30, 8), soprano_three_musicmaker_one],
1161         [(30, 8), (32, 8), soprano_three_musicmaker_three],
1162         [(32, 8), (34, 8), soprano_three_musicmaker_three],

```

```

1163 [(34, 8), (36, 8), soprano_three_musicmaker_three],
1164 [(36, 8), (38, 8), soprano_three_musicmaker_three],
1165 [(38, 8), (40, 8), soprano_three_musicmaker_one],
1166 [(40, 8), (42, 8), soprano_three_musicmaker_three],
1167 [(42, 8), (44, 8), soprano_three_musicmaker_three],
1168 [(44, 8), (46, 8), soprano_three_musicmaker_three],
1169 [(46, 8), (48, 8), soprano_three_musicmaker_one],
1170 [(48, 8), (50, 8), soprano_three_musicmaker_three],
1171 [(50, 8), (52, 8), soprano_three_musicmaker_three],
1172 [(52, 8), (54, 8), soprano_three_musicmaker_one],
1173 [(54, 8), (56, 8), soprano_three_musicmaker_one],
1174 [(56, 8), (58, 8), soprano_three_musicmaker_three],
1175 [(58, 8), (60, 8), soprano_three_musicmaker_three],
1176 [(60, 8), (62, 8), soprano_three_musicmaker_one],
1177 [(62, 8), (64, 8), soprano_three_musicmaker_one],
1178 [(64, 8), (66, 8), soprano_three_musicmaker_three],
1179 [(66, 8), (68, 8), soprano_three_musicmaker_three],
1180 [(68, 8), (70, 8), soprano_three_musicmaker_three],
1181 [(70, 8), (72, 8), soprano_three_musicmaker_two],
1182 [(72, 8), (74, 8), soprano_three_musicmaker_two],
1183 [(74, 8), (76, 8), soprano_three_musicmaker_two],
1184 [(76, 8), (78, 8), soprano_three_musicmaker_two],
1185 [(78, 8), (80, 8), soprano_three_musicmaker_two],
1186 [(80, 8), (82, 8), soprano_three_musicmaker_two],
1187 [(82, 8), (84, 8), soprano_three_musicmaker_two],
1188 [(84, 8), (86, 8), soprano_three_musicmaker_two],
1189 [(86, 8), (88, 8), soprano_three_musicmaker_two],
1190 [(88, 8), (90, 8), soprano_three_musicmaker_one],
1191 [(90, 8), (92, 8), soprano_three_musicmaker_three],
1192 [(92, 8), (94, 8), soprano_three_musicmaker_three],
1193 [(94, 8), (96, 8), soprano_three_musicmaker_three],
1194 [(96, 8), (98, 8), soprano_three_musicmaker_three],
1195 [(98, 8), (100, 8), soprano_three_musicmaker_three],
1196 [(100, 8), (102, 8), soprano_three_musicmaker_three],
1197 [(102, 8), (104, 8), soprano_three_musicmaker_three],
1198 [(104, 8), (106, 8), soprano_three_musicmaker_one],
1199 [(106, 8), (108, 8), soprano_three_musicmaker_one],
1200 [(108, 8), (110, 8), soprano_three_musicmaker_one],
1201 [(110, 8), (112, 8), soprano_three_musicmaker_three],
1202 [(112, 8), (114, 8), soprano_three_musicmaker_three],
1203 [(114, 8), (116, 8), soprano_three_musicmaker_three],
1204 [(116, 8), (118, 8), soprano_three_musicmaker_one],
1205 [(118, 8), (120, 8), soprano_three_musicmaker_one],
1206 [(120, 8), (122, 8), soprano_three_musicmaker_two],
1207 [(122, 8), (124, 8), soprano_three_musicmaker_two],
1208 [(124, 8), (126, 8), soprano_three_musicmaker_two],
1209 [(126, 8), (128, 8), soprano_three_musicmaker_three],
1210 [(128, 8), (130, 8), soprano_three_musicmaker_three],
1211 [(130, 8), (132, 8), soprano_three_musicmaker_one],
1212 [(132, 8), (134, 8), soprano_three_musicmaker_one],
1213 [(134, 8), (136, 8), soprano_three_musicmaker_one],
1214 [(136, 8), (138, 8), soprano_three_musicmaker_three],
1215 [(138, 8), (140, 8), soprano_three_musicmaker_three],
1216 [(140, 8), (142, 8), soprano_three_musicmaker_one],
1217 [(142, 8), (144, 8), soprano_three_musicmaker_three],
1218 [(144, 8), (146, 8), soprano_three_musicmaker_three],
1219 [(146, 8), (148, 8), soprano_three_musicmaker_three],
1220 [(148, 8), (150, 8), soprano_three_musicmaker_two],
1221 [(150, 8), (152, 8), soprano_three_musicmaker_two],

```

```

1222     [(152, 8), (154, 8), soprano_three_musicmaker_two],
1223     [(154, 8), (156, 8), soprano_three_musicmaker_one],
1224     [(156, 8), (158, 8), soprano_three_musicmaker_one],
1225     [(158, 8), (160, 8), soprano_three_musicmaker_three],
1226     [(160, 8), (162, 8), soprano_three_musicmaker_three],
1227     [(162, 8), (164, 8), soprano_three_musicmaker_three],
1228     [(164, 8), (166, 8), soprano_three_musicmaker_three],
1229     [(166, 8), (168, 8), soprano_three_musicmaker_one],
1230     [(168, 8), (170, 8), soprano_three_musicmaker_three],
1231     [(170, 8), (172, 8), soprano_three_musicmaker_three],
1232     [(172, 8), (174, 8), soprano_three_musicmaker_three],
1233     [(174, 8), (176, 8), soprano_three_musicmaker_three],
1234     [(176, 8), (178, 8), soprano_three_musicmaker_three],
1235     [(178, 8), (180, 8), soprano_three_musicmaker_three],
1236     [(180, 8), (182, 8), soprano_three_musicmaker_two],
1237     [(182, 8), (184, 8), soprano_three_musicmaker_two],
1238     [(184, 8), (186, 8), soprano_three_musicmaker_two],
1239     [(186, 8), (188, 8), soprano_three_musicmaker_one],
1240     [(188, 8), (190, 8), soprano_three_musicmaker_three],
1241     [(190, 8), (192, 8), soprano_three_musicmaker_one],
1242     [(192, 8), (194, 8), soprano_three_musicmaker_one],
1243     [(194, 8), (196, 8), soprano_three_musicmaker_three],
1244     [(196, 8), (198, 8), soprano_three_musicmaker_one],
1245     [(198, 8), (199, 8), soprano_three_musicmaker_two],
1246     [(199, 8), (200, 8), soprano_three_musicmaker_two],
1247 ]
1248 ])
1249
1250 voice_5_timespan_list = abjad.TimespanList([
1251     abjad.AnnotatedTimespan(
1252         start_offset=start_offset,
1253         stop_offset=stop_offset,
1254         annotation=MusicSpecifier(
1255             music_maker=music_maker,
1256             voice_name='Voice 5',
1257         ),
1258     ),
1259     for start_offset, stop_offset, music_maker in [
1260         [(0, 8), (2, 8), alto_one_musicmaker_three],
1261         [(2, 8), (4, 8), alto_one_musicmaker_three],
1262         [(4, 8), (6, 8), alto_one_musicmaker_one],
1263         [(6, 8), (8, 8), alto_one_musicmaker_one],
1264         [(8, 8), (10, 8), alto_one_musicmaker_one],
1265         [(10, 8), (12, 8), alto_one_musicmaker_three],
1266         [(12, 8), (14, 8), alto_one_musicmaker_one],
1267         [(14, 8), (16, 8), alto_one_musicmaker_three],
1268         [(16, 8), (18, 8), alto_one_musicmaker_three],
1269         [(18, 8), (20, 8), alto_one_musicmaker_three],
1270         [(20, 8), (22, 8), alto_one_musicmaker_three],
1271         [(22, 8), (24, 8), alto_one_musicmaker_one],
1272         [(24, 8), (26, 8), alto_one_musicmaker_three],
1273         [(26, 8), (28, 8), alto_one_musicmaker_one],
1274         [(28, 8), (30, 8), alto_one_musicmaker_one],
1275         [(30, 8), (32, 8), alto_one_musicmaker_two],
1276         [(32, 8), (34, 8), alto_one_musicmaker_two],
1277         [(34, 8), (36, 8), alto_one_musicmaker_two],
1278         [(36, 8), (38, 8), alto_one_musicmaker_one],
1279         [(38, 8), (40, 8), alto_one_musicmaker_three],
1280         [(40, 8), (42, 8), alto_one_musicmaker_three],

```

```
1281 [(42, 8), (44, 8), alto_one_musicmaker_three],  
1282 [(44, 8), (46, 8), alto_one_musicmaker_three],  
1283 [(46, 8), (48, 8), alto_one_musicmaker_two],  
1284 [(48, 8), (50, 8), alto_one_musicmaker_two],  
1285 [(50, 8), (52, 8), alto_one_musicmaker_two],  
1286 [(52, 8), (54, 8), alto_one_musicmaker_two],  
1287 [(54, 8), (56, 8), alto_one_musicmaker_three],  
1288 [(56, 8), (58, 8), alto_one_musicmaker_three],  
1289 [(58, 8), (60, 8), alto_one_musicmaker_three],  
1290 [(60, 8), (62, 8), alto_one_musicmaker_three],  
1291 [(62, 8), (64, 8), alto_one_musicmaker_three],  
1292 [(64, 8), (66, 8), alto_one_musicmaker_three],  
1293 [(66, 8), (68, 8), alto_one_musicmaker_three],  
1294 [(68, 8), (70, 8), alto_one_musicmaker_three],  
1295 [(70, 8), (72, 8), alto_one_musicmaker_two],  
1296 [(72, 8), (74, 8), alto_one_musicmaker_one],  
1297 [(74, 8), (76, 8), alto_one_musicmaker_one],  
1298 [(76, 8), (78, 8), alto_one_musicmaker_two],  
1299 [(78, 8), (80, 8), alto_one_musicmaker_two],  
1300 [(80, 8), (82, 8), alto_one_musicmaker_three],  
1301 [(82, 8), (84, 8), alto_one_musicmaker_three],  
1302 [(84, 8), (86, 8), alto_one_musicmaker_two],  
1303 [(86, 8), (88, 8), alto_one_musicmaker_one],  
1304 [(88, 8), (90, 8), alto_one_musicmaker_two],  
1305 [(90, 8), (92, 8), alto_one_musicmaker_three],  
1306 [(92, 8), (94, 8), alto_one_musicmaker_one],  
1307 [(94, 8), (96, 8), alto_one_musicmaker_two],  
1308 [(96, 8), (98, 8), alto_one_musicmaker_two],  
1309 [(98, 8), (100, 8), alto_one_musicmaker_two],  
1310 [(100, 8), (102, 8), alto_one_musicmaker_three],  
1311 [(102, 8), (104, 8), alto_one_musicmaker_three],  
1312 [(104, 8), (106, 8), alto_one_musicmaker_three],  
1313 [(106, 8), (108, 8), alto_one_musicmaker_two],  
1314 [(108, 8), (110, 8), alto_one_musicmaker_one],  
1315 [(110, 8), (112, 8), alto_one_musicmaker_two],  
1316 [(112, 8), (114, 8), alto_one_musicmaker_three],  
1317 [(114, 8), (116, 8), alto_one_musicmaker_three],  
1318 [(116, 8), (118, 8), alto_one_musicmaker_three],  
1319 [(118, 8), (120, 8), alto_one_musicmaker_one],  
1320 [(120, 8), (122, 8), alto_one_musicmaker_one],  
1321 [(122, 8), (124, 8), alto_one_musicmaker_three],  
1322 [(124, 8), (126, 8), alto_one_musicmaker_three],  
1323 [(126, 8), (128, 8), alto_one_musicmaker_one],  
1324 [(128, 8), (130, 8), alto_one_musicmaker_two],  
1325 [(130, 8), (132, 8), alto_one_musicmaker_two],  
1326 [(132, 8), (134, 8), alto_one_musicmaker_two],  
1327 [(134, 8), (136, 8), alto_one_musicmaker_two],  
1328 [(136, 8), (138, 8), alto_one_musicmaker_three],  
1329 [(138, 8), (140, 8), alto_one_musicmaker_three],  
1330 [(140, 8), (142, 8), alto_one_musicmaker_three],  
1331 [(142, 8), (144, 8), alto_one_musicmaker_three],  
1332 [(144, 8), (146, 8), alto_one_musicmaker_three],  
1333 [(146, 8), (148, 8), alto_one_musicmaker_two],  
1334 [(148, 8), (150, 8), alto_one_musicmaker_two],  
1335 [(150, 8), (152, 8), alto_one_musicmaker_one],  
1336 [(152, 8), (154, 8), alto_one_musicmaker_two],  
1337 [(154, 8), (156, 8), alto_one_musicmaker_three],  
1338 [(156, 8), (158, 8), alto_one_musicmaker_three],  
1339 [(158, 8), (160, 8), alto_one_musicmaker_three],
```

```

1340     [(160, 8), (162, 8), alto_one_musicmaker_three],
1341     [(162, 8), (164, 8), alto_one_musicmaker_three],
1342     [(164, 8), (166, 8), alto_one_musicmaker_three],
1343     [(166, 8), (168, 8), alto_one_musicmaker_three],
1344     [(168, 8), (170, 8), alto_one_musicmaker_one],
1345     [(170, 8), (172, 8), alto_one_musicmaker_one],
1346     [(172, 8), (174, 8), alto_one_musicmaker_three],
1347     [(174, 8), (176, 8), alto_one_musicmaker_one],
1348     [(176, 8), (178, 8), alto_one_musicmaker_three],
1349     [(178, 8), (180, 8), alto_one_musicmaker_three],
1350     [(180, 8), (182, 8), alto_one_musicmaker_one],
1351     [(182, 8), (184, 8), alto_one_musicmaker_three],
1352     [(184, 8), (186, 8), alto_one_musicmaker_three],
1353     [(186, 8), (188, 8), alto_one_musicmaker_one],
1354     [(188, 8), (190, 8), alto_one_musicmaker_three],
1355     [(190, 8), (192, 8), alto_one_musicmaker_three],
1356     [(192, 8), (194, 8), alto_one_musicmaker_one],
1357     [(194, 8), (196, 8), alto_one_musicmaker_two],
1358     [(196, 8), (198, 8), alto_one_musicmaker_two],
1359     [(198, 8), (199, 8), alto_one_musicmaker_one],
1360     [(199, 8), (200, 8), alto_one_musicmaker_two],
1361   ],
1362 ]
1363
1364 voice_6_timespan_list = abjad.TimespanList([
1365     abjad.AnnotatedTimespan(
1366         start_offset=start_offset,
1367         stop_offset=stop_offset,
1368         annotation=MusicSpecifier(
1369             music_maker=music_maker,
1370             voice_name='Voice 6',
1371         ),
1372     ),
1373     for start_offset, stop_offset, music_maker in [
1374         [(0, 8), (2, 8), alto_two_musicmaker_three],
1375         [(2, 8), (4, 8), alto_two_musicmaker_three],
1376         [(4, 8), (6, 8), alto_two_musicmaker_one],
1377         [(6, 8), (8, 8), alto_two_musicmaker_two],
1378         [(8, 8), (10, 8), alto_two_musicmaker_two],
1379         [(10, 8), (12, 8), alto_two_musicmaker_one],
1380         [(12, 8), (14, 8), alto_two_musicmaker_two],
1381         [(14, 8), (16, 8), alto_two_musicmaker_one],
1382         [(16, 8), (18, 8), alto_two_musicmaker_one],
1383         [(18, 8), (20, 8), alto_two_musicmaker_three],
1384         [(20, 8), (22, 8), alto_two_musicmaker_three],
1385         [(22, 8), (24, 8), alto_two_musicmaker_one],
1386         [(24, 8), (26, 8), alto_two_musicmaker_one],
1387         [(26, 8), (28, 8), alto_two_musicmaker_three],
1388         [(28, 8), (30, 8), alto_two_musicmaker_three],
1389         [(30, 8), (32, 8), alto_two_musicmaker_two],
1390         [(32, 8), (34, 8), alto_two_musicmaker_two],
1391         [(34, 8), (36, 8), alto_two_musicmaker_two],
1392         [(36, 8), (38, 8), alto_two_musicmaker_one],
1393         [(38, 8), (40, 8), alto_two_musicmaker_one],
1394         [(40, 8), (42, 8), alto_two_musicmaker_three],
1395         [(42, 8), (44, 8), alto_two_musicmaker_three],
1396         [(44, 8), (46, 8), alto_two_musicmaker_three],
1397         [(46, 8), (48, 8), alto_two_musicmaker_three],
1398         [(48, 8), (50, 8), alto_two_musicmaker_three],

```

```
1399 [(50, 8), (52, 8), alto_two_musicmaker_two],  
1400 [(52, 8), (54, 8), alto_two_musicmaker_two],  
1401 [(54, 8), (56, 8), alto_two_musicmaker_one],  
1402 [(56, 8), (58, 8), alto_two_musicmaker_one],  
1403 [(58, 8), (60, 8), alto_two_musicmaker_two],  
1404 [(60, 8), (62, 8), alto_two_musicmaker_two],  
1405 [(62, 8), (64, 8), alto_two_musicmaker_three],  
1406 [(64, 8), (66, 8), alto_two_musicmaker_three],  
1407 [(66, 8), (68, 8), alto_two_musicmaker_three],  
1408 [(68, 8), (70, 8), alto_two_musicmaker_three],  
1409 [(70, 8), (72, 8), alto_two_musicmaker_two],  
1410 [(72, 8), (74, 8), alto_two_musicmaker_one],  
1411 [(74, 8), (76, 8), alto_two_musicmaker_two],  
1412 [(76, 8), (78, 8), alto_two_musicmaker_three],  
1413 [(78, 8), (80, 8), alto_two_musicmaker_three],  
1414 [(80, 8), (82, 8), alto_two_musicmaker_one],  
1415 [(82, 8), (84, 8), alto_two_musicmaker_one],  
1416 [(84, 8), (86, 8), alto_two_musicmaker_two],  
1417 [(86, 8), (88, 8), alto_two_musicmaker_two],  
1418 [(88, 8), (90, 8), alto_two_musicmaker_two],  
1419 [(90, 8), (92, 8), alto_two_musicmaker_three],  
1420 [(92, 8), (94, 8), alto_two_musicmaker_three],  
1421 [(94, 8), (96, 8), alto_two_musicmaker_three],  
1422 [(96, 8), (98, 8), alto_two_musicmaker_three],  
1423 [(98, 8), (100, 8), alto_two_musicmaker_one],  
1424 [(100, 8), (102, 8), alto_two_musicmaker_one],  
1425 [(102, 8), (104, 8), alto_two_musicmaker_one],  
1426 [(104, 8), (106, 8), alto_two_musicmaker_three],  
1427 [(106, 8), (108, 8), alto_two_musicmaker_three],  
1428 [(108, 8), (110, 8), alto_two_musicmaker_three],  
1429 [(110, 8), (112, 8), alto_two_musicmaker_three],  
1430 [(112, 8), (114, 8), alto_two_musicmaker_two],  
1431 [(114, 8), (116, 8), alto_two_musicmaker_two],  
1432 [(116, 8), (118, 8), alto_two_musicmaker_two],  
1433 [(118, 8), (120, 8), alto_two_musicmaker_two],  
1434 [(120, 8), (122, 8), alto_two_musicmaker_two],  
1435 [(122, 8), (124, 8), alto_two_musicmaker_one],  
1436 [(124, 8), (126, 8), alto_two_musicmaker_one],  
1437 [(126, 8), (128, 8), alto_two_musicmaker_three],  
1438 [(128, 8), (130, 8), alto_two_musicmaker_three],  
1439 [(130, 8), (132, 8), alto_two_musicmaker_two],  
1440 [(132, 8), (134, 8), alto_two_musicmaker_two],  
1441 [(134, 8), (136, 8), alto_two_musicmaker_two],  
1442 [(136, 8), (138, 8), alto_two_musicmaker_three],  
1443 [(138, 8), (140, 8), alto_two_musicmaker_three],  
1444 [(140, 8), (142, 8), alto_two_musicmaker_one],  
1445 [(142, 8), (144, 8), alto_two_musicmaker_one],  
1446 [(144, 8), (146, 8), alto_two_musicmaker_three],  
1447 [(146, 8), (148, 8), alto_two_musicmaker_three],  
1448 [(148, 8), (150, 8), alto_two_musicmaker_three],  
1449 [(150, 8), (152, 8), alto_two_musicmaker_three],  
1450 [(152, 8), (154, 8), alto_two_musicmaker_three],  
1451 [(154, 8), (156, 8), alto_two_musicmaker_three],  
1452 [(156, 8), (158, 8), alto_two_musicmaker_three],  
1453 [(158, 8), (160, 8), alto_two_musicmaker_two],  
1454 [(160, 8), (162, 8), alto_two_musicmaker_two],  
1455 [(162, 8), (164, 8), alto_two_musicmaker_two],  
1456 [(164, 8), (166, 8), alto_two_musicmaker_one],  
1457 [(166, 8), (168, 8), alto_two_musicmaker_one],
```

```

1458     [(168, 8), (170, 8), alto_two_musicmaker_two],
1459     [(170, 8), (172, 8), alto_two_musicmaker_one],
1460     [(172, 8), (174, 8), alto_two_musicmaker_one],
1461     [(174, 8), (176, 8), alto_two_musicmaker_one],
1462     [(176, 8), (178, 8), alto_two_musicmaker_three],
1463     [(178, 8), (180, 8), alto_two_musicmaker_three],
1464     [(180, 8), (182, 8), alto_two_musicmaker_three],
1465     [(182, 8), (184, 8), alto_two_musicmaker_three],
1466     [(184, 8), (186, 8), alto_two_musicmaker_three],
1467     [(186, 8), (188, 8), alto_two_musicmaker_two],
1468     [(188, 8), (190, 8), alto_two_musicmaker_one],
1469     [(190, 8), (192, 8), alto_two_musicmaker_one],
1470     [(192, 8), (194, 8), alto_two_musicmaker_two],
1471     [(194, 8), (196, 8), alto_two_musicmaker_three],
1472     [(196, 8), (198, 8), alto_two_musicmaker_three],
1473     [(198, 8), (199, 8), alto_two_musicmaker_three],
1474     [(199, 8), (200, 8), alto_two_musicmaker_three],
1475 ]
1476 ])
1477
1478 voice_7_timespan_list = abjad.TimespanList([
1479     abjad.AnnotatedTimespan(
1480         start_offset=start_offset,
1481         stop_offset=stop_offset,
1482         annotation=MusicSpecifier(
1483             music_maker=music_maker,
1484             voice_name='Voice 7',
1485         ),
1486     )
1487     for start_offset, stop_offset, music_maker in [
1488         [(0, 8), (2, 8), alto_three_musicmaker_three],
1489         [(2, 8), (4, 8), alto_three_musicmaker_three],
1490         [(4, 8), (6, 8), alto_three_musicmaker_three],
1491         [(6, 8), (8, 8), alto_three_musicmaker_one],
1492         [(8, 8), (10, 8), alto_three_musicmaker_two],
1493         [(10, 8), (12, 8), alto_three_musicmaker_two],
1494         [(12, 8), (14, 8), alto_three_musicmaker_two],
1495         [(14, 8), (16, 8), alto_three_musicmaker_three],
1496         [(16, 8), (18, 8), alto_three_musicmaker_three],
1497         [(18, 8), (20, 8), alto_three_musicmaker_three],
1498         [(20, 8), (22, 8), alto_three_musicmaker_three],
1499         [(22, 8), (24, 8), alto_three_musicmaker_one],
1500         [(24, 8), (26, 8), alto_three_musicmaker_three],
1501         [(26, 8), (28, 8), alto_three_musicmaker_one],
1502         [(28, 8), (30, 8), alto_three_musicmaker_one],
1503         [(30, 8), (32, 8), alto_three_musicmaker_one],
1504         [(32, 8), (34, 8), alto_three_musicmaker_two],
1505         [(34, 8), (36, 8), alto_three_musicmaker_two],
1506         [(36, 8), (38, 8), alto_three_musicmaker_three],
1507         [(38, 8), (40, 8), alto_three_musicmaker_three],
1508         [(40, 8), (42, 8), alto_three_musicmaker_three],
1509         [(42, 8), (44, 8), alto_three_musicmaker_three],
1510         [(44, 8), (46, 8), alto_three_musicmaker_three],
1511         [(46, 8), (48, 8), alto_three_musicmaker_two],
1512         [(48, 8), (50, 8), alto_three_musicmaker_one],
1513         [(50, 8), (52, 8), alto_three_musicmaker_two],
1514         [(52, 8), (54, 8), alto_three_musicmaker_one],
1515         [(54, 8), (56, 8), alto_three_musicmaker_three],
1516         [(56, 8), (58, 8), alto_three_musicmaker_three],

```

```
1517 [(58, 8), (60, 8), alto_three_musicmaker_three],  
1518 [(60, 8), (62, 8), alto_three_musicmaker_one],  
1519 [(62, 8), (64, 8), alto_three_musicmaker_three],  
1520 [(64, 8), (66, 8), alto_three_musicmaker_three],  
1521 [(66, 8), (68, 8), alto_three_musicmaker_three],  
1522 [(68, 8), (70, 8), alto_three_musicmaker_one],  
1523 [(70, 8), (72, 8), alto_three_musicmaker_one],  
1524 [(72, 8), (74, 8), alto_three_musicmaker_one],  
1525 [(74, 8), (76, 8), alto_three_musicmaker_three],  
1526 [(76, 8), (78, 8), alto_three_musicmaker_three],  
1527 [(78, 8), (80, 8), alto_three_musicmaker_one],  
1528 [(80, 8), (82, 8), alto_three_musicmaker_two],  
1529 [(82, 8), (84, 8), alto_three_musicmaker_two],  
1530 [(84, 8), (86, 8), alto_three_musicmaker_three],  
1531 [(86, 8), (88, 8), alto_three_musicmaker_three],  
1532 [(88, 8), (90, 8), alto_three_musicmaker_one],  
1533 [(90, 8), (92, 8), alto_three_musicmaker_three],  
1534 [(92, 8), (94, 8), alto_three_musicmaker_one],  
1535 [(94, 8), (96, 8), alto_three_musicmaker_one],  
1536 [(96, 8), (98, 8), alto_three_musicmaker_three],  
1537 [(98, 8), (100, 8), alto_three_musicmaker_three],  
1538 [(100, 8), (102, 8), alto_three_musicmaker_three],  
1539 [(102, 8), (104, 8), alto_three_musicmaker_three],  
1540 [(104, 8), (106, 8), alto_three_musicmaker_three],  
1541 [(106, 8), (108, 8), alto_three_musicmaker_three],  
1542 [(108, 8), (110, 8), alto_three_musicmaker_two],  
1543 [(110, 8), (112, 8), alto_three_musicmaker_one],  
1544 [(112, 8), (114, 8), alto_three_musicmaker_one],  
1545 [(114, 8), (116, 8), alto_three_musicmaker_two],  
1546 [(116, 8), (118, 8), alto_three_musicmaker_three],  
1547 [(118, 8), (120, 8), alto_three_musicmaker_three],  
1548 [(120, 8), (122, 8), alto_three_musicmaker_one],  
1549 [(122, 8), (124, 8), alto_three_musicmaker_one],  
1550 [(124, 8), (126, 8), alto_three_musicmaker_three],  
1551 [(126, 8), (128, 8), alto_three_musicmaker_three],  
1552 [(128, 8), (130, 8), alto_three_musicmaker_one],  
1553 [(130, 8), (132, 8), alto_three_musicmaker_one],  
1554 [(132, 8), (134, 8), alto_three_musicmaker_three],  
1555 [(134, 8), (136, 8), alto_three_musicmaker_three],  
1556 [(136, 8), (138, 8), alto_three_musicmaker_one],  
1557 [(138, 8), (140, 8), alto_three_musicmaker_one],  
1558 [(140, 8), (142, 8), alto_three_musicmaker_three],  
1559 [(142, 8), (144, 8), alto_three_musicmaker_three],  
1560 [(144, 8), (146, 8), alto_three_musicmaker_two],  
1561 [(146, 8), (148, 8), alto_three_musicmaker_one],  
1562 [(148, 8), (150, 8), alto_three_musicmaker_two],  
1563 [(150, 8), (152, 8), alto_three_musicmaker_one],  
1564 [(152, 8), (154, 8), alto_three_musicmaker_one],  
1565 [(154, 8), (156, 8), alto_three_musicmaker_one],  
1566 [(156, 8), (158, 8), alto_three_musicmaker_three],  
1567 [(158, 8), (160, 8), alto_three_musicmaker_three],  
1568 [(160, 8), (162, 8), alto_three_musicmaker_three],  
1569 [(162, 8), (164, 8), alto_three_musicmaker_two],  
1570 [(164, 8), (166, 8), alto_three_musicmaker_one],  
1571 [(166, 8), (168, 8), alto_three_musicmaker_one],  
1572 [(168, 8), (170, 8), alto_three_musicmaker_two],  
1573 [(170, 8), (172, 8), alto_three_musicmaker_two],  
1574 [(172, 8), (174, 8), alto_three_musicmaker_three],  
1575 [(174, 8), (176, 8), alto_three_musicmaker_three],
```

```

1576     [(176, 8), (178, 8), alto_three_musicmaker_one],
1577     [(178, 8), (180, 8), alto_three_musicmaker_three],
1578     [(180, 8), (182, 8), alto_three_musicmaker_one],
1579     [(182, 8), (184, 8), alto_three_musicmaker_one],
1580     [(184, 8), (186, 8), alto_three_musicmaker_three],
1581     [(186, 8), (188, 8), alto_three_musicmaker_three],
1582     [(188, 8), (190, 8), alto_three_musicmaker_two],
1583     [(190, 8), (192, 8), alto_three_musicmaker_one],
1584     [(192, 8), (194, 8), alto_three_musicmaker_two],
1585     [(194, 8), (196, 8), alto_three_musicmaker_two],
1586     [(196, 8), (198, 8), alto_three_musicmaker_three],
1587     [(198, 8), (199, 8), alto_three_musicmaker_three],
1588     [(199, 8), (200, 8), alto_three_musicmaker_three],
1589   ]
1590 ])
1591
1592 voice_8_timespan_list = abjad.TimespanList([
1593     abjad.AnnotatedTimespan(
1594         start_offset=start_offset,
1595         stop_offset=stop_offset,
1596         annotation=MusicSpecifier(
1597             music_maker=music_maker,
1598             voice_name='Voice 8',
1599         ),
1600     )
1601     for start_offset, stop_offset, music_maker in [
1602         [(0, 8), (2, 8), alto_four_musicmaker_one],
1603         [(2, 8), (4, 8), alto_four_musicmaker_one],
1604         [(4, 8), (6, 8), alto_four_musicmaker_one],
1605         [(6, 8), (8, 8), alto_four_musicmaker_one],
1606         [(8, 8), (10, 8), alto_four_musicmaker_three],
1607         [(10, 8), (12, 8), alto_four_musicmaker_three],
1608         [(12, 8), (14, 8), alto_four_musicmaker_three],
1609         [(14, 8), (16, 8), alto_four_musicmaker_one],
1610         [(16, 8), (18, 8), alto_four_musicmaker_two],
1611         [(18, 8), (20, 8), alto_four_musicmaker_three],
1612         [(20, 8), (22, 8), alto_four_musicmaker_three],
1613         [(22, 8), (24, 8), alto_four_musicmaker_one],
1614         [(24, 8), (26, 8), alto_four_musicmaker_one],
1615         [(26, 8), (28, 8), alto_four_musicmaker_one],
1616         [(28, 8), (30, 8), alto_four_musicmaker_three],
1617         [(30, 8), (32, 8), alto_four_musicmaker_three],
1618         [(32, 8), (34, 8), alto_four_musicmaker_three],
1619         [(34, 8), (36, 8), alto_four_musicmaker_one],
1620         [(36, 8), (38, 8), alto_four_musicmaker_one],
1621         [(38, 8), (40, 8), alto_four_musicmaker_three],
1622         [(40, 8), (42, 8), alto_four_musicmaker_three],
1623         [(42, 8), (44, 8), alto_four_musicmaker_two],
1624         [(44, 8), (46, 8), alto_four_musicmaker_two],
1625         [(46, 8), (48, 8), alto_four_musicmaker_three],
1626         [(48, 8), (50, 8), alto_four_musicmaker_three],
1627         [(50, 8), (52, 8), alto_four_musicmaker_one],
1628         [(52, 8), (54, 8), alto_four_musicmaker_one],
1629         [(54, 8), (56, 8), alto_four_musicmaker_one],
1630         [(56, 8), (58, 8), alto_four_musicmaker_two],
1631         [(58, 8), (60, 8), alto_four_musicmaker_two],
1632         [(60, 8), (62, 8), alto_four_musicmaker_two],
1633         [(62, 8), (64, 8), alto_four_musicmaker_two],
1634         [(64, 8), (66, 8), alto_four_musicmaker_three],

```

```
1635 [(66, 8), (68, 8), alto_four_musicmaker_three],  
1636 [(68, 8), (70, 8), alto_four_musicmaker_three],  
1637 [(70, 8), (72, 8), alto_four_musicmaker_three],  
1638 [(72, 8), (74, 8), alto_four_musicmaker_one],  
1639 [(74, 8), (76, 8), alto_four_musicmaker_three],  
1640 [(76, 8), (78, 8), alto_four_musicmaker_three],  
1641 [(78, 8), (80, 8), alto_four_musicmaker_three],  
1642 [(80, 8), (82, 8), alto_four_musicmaker_one],  
1643 [(82, 8), (84, 8), alto_four_musicmaker_one],  
1644 [(84, 8), (86, 8), alto_four_musicmaker_three],  
1645 [(86, 8), (88, 8), alto_four_musicmaker_three],  
1646 [(88, 8), (90, 8), alto_four_musicmaker_three],  
1647 [(90, 8), (92, 8), alto_four_musicmaker_two],  
1648 [(92, 8), (94, 8), alto_four_musicmaker_one],  
1649 [(94, 8), (96, 8), alto_four_musicmaker_two],  
1650 [(96, 8), (98, 8), alto_four_musicmaker_one],  
1651 [(98, 8), (100, 8), alto_four_musicmaker_one],  
1652 [(100, 8), (102, 8), alto_four_musicmaker_one],  
1653 [(102, 8), (104, 8), alto_four_musicmaker_three],  
1654 [(104, 8), (106, 8), alto_four_musicmaker_three],  
1655 [(106, 8), (108, 8), alto_four_musicmaker_three],  
1656 [(108, 8), (110, 8), alto_four_musicmaker_three],  
1657 [(110, 8), (112, 8), alto_four_musicmaker_two],  
1658 [(112, 8), (114, 8), alto_four_musicmaker_one],  
1659 [(114, 8), (116, 8), alto_four_musicmaker_one],  
1660 [(116, 8), (118, 8), alto_four_musicmaker_three],  
1661 [(118, 8), (120, 8), alto_four_musicmaker_three],  
1662 [(120, 8), (122, 8), alto_four_musicmaker_one],  
1663 [(122, 8), (124, 8), alto_four_musicmaker_one],  
1664 [(124, 8), (126, 8), alto_four_musicmaker_three],  
1665 [(126, 8), (128, 8), alto_four_musicmaker_three],  
1666 [(128, 8), (130, 8), alto_four_musicmaker_one],  
1667 [(130, 8), (132, 8), alto_four_musicmaker_three],  
1668 [(132, 8), (134, 8), alto_four_musicmaker_one],  
1669 [(134, 8), (136, 8), alto_four_musicmaker_three],  
1670 [(136, 8), (138, 8), alto_four_musicmaker_two],  
1671 [(138, 8), (140, 8), alto_four_musicmaker_two],  
1672 [(140, 8), (142, 8), alto_four_musicmaker_three],  
1673 [(142, 8), (144, 8), alto_four_musicmaker_one],  
1674 [(144, 8), (146, 8), alto_four_musicmaker_three],  
1675 [(146, 8), (148, 8), alto_four_musicmaker_three],  
1676 [(148, 8), (150, 8), alto_four_musicmaker_one],  
1677 [(150, 8), (152, 8), alto_four_musicmaker_one],  
1678 [(152, 8), (154, 8), alto_four_musicmaker_three],  
1679 [(154, 8), (156, 8), alto_four_musicmaker_one],  
1680 [(156, 8), (158, 8), alto_four_musicmaker_one],  
1681 [(158, 8), (160, 8), alto_four_musicmaker_two],  
1682 [(160, 8), (162, 8), alto_four_musicmaker_three],  
1683 [(162, 8), (164, 8), alto_four_musicmaker_three],  
1684 [(164, 8), (166, 8), alto_four_musicmaker_three],  
1685 [(166, 8), (168, 8), alto_four_musicmaker_three],  
1686 [(168, 8), (170, 8), alto_four_musicmaker_one],  
1687 [(170, 8), (172, 8), alto_four_musicmaker_one],  
1688 [(172, 8), (174, 8), alto_four_musicmaker_one],  
1689 [(174, 8), (176, 8), alto_four_musicmaker_three],  
1690 [(176, 8), (178, 8), alto_four_musicmaker_three],  
1691 [(178, 8), (180, 8), alto_four_musicmaker_three],  
1692 [(180, 8), (182, 8), alto_four_musicmaker_one],  
1693 [(182, 8), (184, 8), alto_four_musicmaker_three],
```

```

1694     [(184, 8), (186, 8), alto_four_musicmaker_one],
1695     [(186, 8), (188, 8), alto_four_musicmaker_three],
1696     [(188, 8), (190, 8), alto_four_musicmaker_one],
1697     [(190, 8), (192, 8), alto_four_musicmaker_one],
1698     [(192, 8), (194, 8), alto_four_musicmaker_three],
1699     [(194, 8), (196, 8), alto_four_musicmaker_one],
1700     [(196, 8), (198, 8), alto_four_musicmaker_two],
1701     [(198, 8), (199, 8), alto_four_musicmaker_three],
1702     [(199, 8), (200, 8), alto_four_musicmaker_three],
1703 ]
1704 ])
1705
1706 voice_9_timespan_list = abjad.TimespanList([
1707     abjad.AnnotatedTimespan(
1708         start_offset=start_offset,
1709         stop_offset=stop_offset,
1710         annotation=MusicSpecifier(
1711             music_maker=music_maker,
1712             voice_name='Voice 9',
1713         ),
1714     )
1715     for start_offset, stop_offset, music_maker in [
1716         [(0, 8), (2, 8), alto_five_musicmaker_three],
1717         [(2, 8), (4, 8), alto_five_musicmaker_three],
1718         [(4, 8), (6, 8), alto_five_musicmaker_three],
1719         [(6, 8), (8, 8), alto_five_musicmaker_one],
1720         [(8, 8), (10, 8), alto_five_musicmaker_three],
1721         [(10, 8), (12, 8), alto_five_musicmaker_one],
1722         [(12, 8), (14, 8), alto_five_musicmaker_three],
1723         [(14, 8), (16, 8), alto_five_musicmaker_three],
1724         [(16, 8), (18, 8), alto_five_musicmaker_three],
1725         [(18, 8), (20, 8), alto_five_musicmaker_three],
1726         [(20, 8), (22, 8), alto_five_musicmaker_two],
1727         [(22, 8), (24, 8), alto_five_musicmaker_two],
1728         [(24, 8), (26, 8), alto_five_musicmaker_one],
1729         [(26, 8), (28, 8), alto_five_musicmaker_one],
1730         [(28, 8), (30, 8), alto_five_musicmaker_two],
1731         [(30, 8), (32, 8), alto_five_musicmaker_three],
1732         [(32, 8), (34, 8), alto_five_musicmaker_three],
1733         [(34, 8), (36, 8), alto_five_musicmaker_three],
1734         [(36, 8), (38, 8), alto_five_musicmaker_three],
1735         [(38, 8), (40, 8), alto_five_musicmaker_three],
1736         [(40, 8), (42, 8), alto_five_musicmaker_three],
1737         [(42, 8), (44, 8), alto_five_musicmaker_three],
1738         [(44, 8), (46, 8), alto_five_musicmaker_one],
1739         [(46, 8), (48, 8), alto_five_musicmaker_three],
1740         [(48, 8), (50, 8), alto_five_musicmaker_one],
1741         [(50, 8), (52, 8), alto_five_musicmaker_three],
1742         [(52, 8), (54, 8), alto_five_musicmaker_one],
1743         [(54, 8), (56, 8), alto_five_musicmaker_two],
1744         [(56, 8), (58, 8), alto_five_musicmaker_two],
1745         [(58, 8), (60, 8), alto_five_musicmaker_three],
1746         [(60, 8), (62, 8), alto_five_musicmaker_three],
1747         [(62, 8), (64, 8), alto_five_musicmaker_three],
1748         [(64, 8), (66, 8), alto_five_musicmaker_three],
1749         [(66, 8), (68, 8), alto_five_musicmaker_three],
1750         [(68, 8), (70, 8), alto_five_musicmaker_three],
1751         [(70, 8), (72, 8), alto_five_musicmaker_three],
1752         [(72, 8), (74, 8), alto_five_musicmaker_two],

```

```
1753 [(74, 8), (76, 8), alto_five_musicmaker_two],  
1754 [(76, 8), (78, 8), alto_five_musicmaker_one],  
1755 [(78, 8), (80, 8), alto_five_musicmaker_three],  
1756 [(80, 8), (82, 8), alto_five_musicmaker_three],  
1757 [(82, 8), (84, 8), alto_five_musicmaker_one],  
1758 [(84, 8), (86, 8), alto_five_musicmaker_one],  
1759 [(86, 8), (88, 8), alto_five_musicmaker_three],  
1760 [(88, 8), (90, 8), alto_five_musicmaker_three],  
1761 [(90, 8), (92, 8), alto_five_musicmaker_three],  
1762 [(92, 8), (94, 8), alto_five_musicmaker_three],  
1763 [(94, 8), (96, 8), alto_five_musicmaker_three],  
1764 [(96, 8), (98, 8), alto_five_musicmaker_three],  
1765 [(98, 8), (100, 8), alto_five_musicmaker_two],  
1766 [(100, 8), (102, 8), alto_five_musicmaker_two],  
1767 [(102, 8), (104, 8), alto_five_musicmaker_two],  
1768 [(104, 8), (106, 8), alto_five_musicmaker_one],  
1769 [(106, 8), (108, 8), alto_five_musicmaker_one],  
1770 [(108, 8), (110, 8), alto_five_musicmaker_one],  
1771 [(110, 8), (112, 8), alto_five_musicmaker_two],  
1772 [(112, 8), (114, 8), alto_five_musicmaker_one],  
1773 [(114, 8), (116, 8), alto_five_musicmaker_three],  
1774 [(116, 8), (118, 8), alto_five_musicmaker_three],  
1775 [(118, 8), (120, 8), alto_five_musicmaker_three],  
1776 [(120, 8), (122, 8), alto_five_musicmaker_three],  
1777 [(122, 8), (124, 8), alto_five_musicmaker_three],  
1778 [(124, 8), (126, 8), alto_five_musicmaker_two],  
1779 [(126, 8), (128, 8), alto_five_musicmaker_one],  
1780 [(128, 8), (130, 8), alto_five_musicmaker_two],  
1781 [(130, 8), (132, 8), alto_five_musicmaker_three],  
1782 [(132, 8), (134, 8), alto_five_musicmaker_three],  
1783 [(134, 8), (136, 8), alto_five_musicmaker_three],  
1784 [(136, 8), (138, 8), alto_five_musicmaker_three],  
1785 [(138, 8), (140, 8), alto_five_musicmaker_one],  
1786 [(140, 8), (142, 8), alto_five_musicmaker_three],  
1787 [(142, 8), (144, 8), alto_five_musicmaker_three],  
1788 [(144, 8), (146, 8), alto_five_musicmaker_three],  
1789 [(146, 8), (148, 8), alto_five_musicmaker_three],  
1790 [(148, 8), (150, 8), alto_five_musicmaker_one],  
1791 [(150, 8), (152, 8), alto_five_musicmaker_two],  
1792 [(152, 8), (154, 8), alto_five_musicmaker_one],  
1793 [(154, 8), (156, 8), alto_five_musicmaker_one],  
1794 [(156, 8), (158, 8), alto_five_musicmaker_one],  
1795 [(158, 8), (160, 8), alto_five_musicmaker_three],  
1796 [(160, 8), (162, 8), alto_five_musicmaker_three],  
1797 [(162, 8), (164, 8), alto_five_musicmaker_three],  
1798 [(164, 8), (166, 8), alto_five_musicmaker_three],  
1799 [(166, 8), (168, 8), alto_five_musicmaker_three],  
1800 [(168, 8), (170, 8), alto_five_musicmaker_three],  
1801 [(170, 8), (172, 8), alto_five_musicmaker_two],  
1802 [(172, 8), (174, 8), alto_five_musicmaker_two],  
1803 [(174, 8), (176, 8), alto_five_musicmaker_two],  
1804 [(176, 8), (178, 8), alto_five_musicmaker_two],  
1805 [(178, 8), (180, 8), alto_five_musicmaker_one],  
1806 [(180, 8), (182, 8), alto_five_musicmaker_one],  
1807 [(182, 8), (184, 8), alto_five_musicmaker_three],  
1808 [(184, 8), (186, 8), alto_five_musicmaker_three],  
1809 [(186, 8), (188, 8), alto_five_musicmaker_three],  
1810 [(188, 8), (190, 8), alto_five_musicmaker_three],  
1811 [(190, 8), (192, 8), alto_five_musicmaker_three],
```

```

1812     [(192, 8), (194, 8), alto_five_musicmaker_three],
1813     [(194, 8), (196, 8), alto_five_musicmaker_two],
1814     [(196, 8), (198, 8), alto_five_musicmaker_one],
1815     [(198, 8), (199, 8), alto_five_musicmaker_one],
1816     [(199, 8), (200, 8), alto_five_musicmaker_two],
1817 ]
1818 ])
1819
1820 voice_10_timespan_list = abjad.TimespanList([
1821     abjad.AnnotatedTimespan(
1822         start_offset=start_offset,
1823         stop_offset=stop_offset,
1824         annotation=MusicSpecifier(
1825             music_maker=music_maker,
1826             voice_name='Voice 10',
1827         ),
1828     )
1829     for start_offset, stop_offset, music_maker in [
1830         [(0, 8), (2, 8), alto_six_musicmaker_three],
1831         [(2, 8), (4, 8), alto_six_musicmaker_three],
1832         [(4, 8), (6, 8), alto_six_musicmaker_two],
1833         [(6, 8), (8, 8), alto_six_musicmaker_two],
1834         [(8, 8), (10, 8), alto_six_musicmaker_one],
1835         [(10, 8), (12, 8), alto_six_musicmaker_one],
1836         [(12, 8), (14, 8), alto_six_musicmaker_one],
1837         [(14, 8), (16, 8), alto_six_musicmaker_two],
1838         [(16, 8), (18, 8), alto_six_musicmaker_three],
1839         [(18, 8), (20, 8), alto_six_musicmaker_three],
1840         [(20, 8), (22, 8), alto_six_musicmaker_three],
1841         [(22, 8), (24, 8), alto_six_musicmaker_three],
1842         [(24, 8), (26, 8), alto_six_musicmaker_three],
1843         [(26, 8), (28, 8), alto_six_musicmaker_three],
1844         [(28, 8), (30, 8), alto_six_musicmaker_one],
1845         [(30, 8), (32, 8), alto_six_musicmaker_one],
1846         [(32, 8), (34, 8), alto_six_musicmaker_two],
1847         [(34, 8), (36, 8), alto_six_musicmaker_two],
1848         [(36, 8), (38, 8), alto_six_musicmaker_two],
1849         [(38, 8), (40, 8), alto_six_musicmaker_two],
1850         [(40, 8), (42, 8), alto_six_musicmaker_two],
1851         [(42, 8), (44, 8), alto_six_musicmaker_three],
1852         [(44, 8), (46, 8), alto_six_musicmaker_three],
1853         [(46, 8), (48, 8), alto_six_musicmaker_three],
1854         [(48, 8), (50, 8), alto_six_musicmaker_three],
1855         [(50, 8), (52, 8), alto_six_musicmaker_three],
1856         [(52, 8), (54, 8), alto_six_musicmaker_three],
1857         [(54, 8), (56, 8), alto_six_musicmaker_three],
1858         [(56, 8), (58, 8), alto_six_musicmaker_two],
1859         [(58, 8), (60, 8), alto_six_musicmaker_one],
1860         [(60, 8), (62, 8), alto_six_musicmaker_one],
1861         [(62, 8), (64, 8), alto_six_musicmaker_two],
1862         [(64, 8), (66, 8), alto_six_musicmaker_one],
1863         [(66, 8), (68, 8), alto_six_musicmaker_three],
1864         [(68, 8), (70, 8), alto_six_musicmaker_three],
1865         [(70, 8), (72, 8), alto_six_musicmaker_three],
1866         [(72, 8), (74, 8), alto_six_musicmaker_three],
1867         [(74, 8), (76, 8), alto_six_musicmaker_three],
1868         [(76, 8), (78, 8), alto_six_musicmaker_three],
1869         [(78, 8), (80, 8), alto_six_musicmaker_three],
1870         [(80, 8), (82, 8), alto_six_musicmaker_two],

```

```
1871 [(82, 8), (84, 8), alto_six_musicmaker_two],  
1872 [(84, 8), (86, 8), alto_six_musicmaker_one],  
1873 [(86, 8), (88, 8), alto_six_musicmaker_three],  
1874 [(88, 8), (90, 8), alto_six_musicmaker_three],  
1875 [(90, 8), (92, 8), alto_six_musicmaker_three],  
1876 [(92, 8), (94, 8), alto_six_musicmaker_three],  
1877 [(94, 8), (96, 8), alto_six_musicmaker_two],  
1878 [(96, 8), (98, 8), alto_six_musicmaker_one],  
1879 [(98, 8), (100, 8), alto_six_musicmaker_one],  
1880 [(100, 8), (102, 8), alto_six_musicmaker_one],  
1881 [(102, 8), (104, 8), alto_six_musicmaker_three],  
1882 [(104, 8), (106, 8), alto_six_musicmaker_three],  
1883 [(106, 8), (108, 8), alto_six_musicmaker_three],  
1884 [(108, 8), (110, 8), alto_six_musicmaker_three],  
1885 [(110, 8), (112, 8), alto_six_musicmaker_three],  
1886 [(112, 8), (114, 8), alto_six_musicmaker_three],  
1887 [(114, 8), (116, 8), alto_six_musicmaker_three],  
1888 [(116, 8), (118, 8), alto_six_musicmaker_two],  
1889 [(118, 8), (120, 8), alto_six_musicmaker_one],  
1890 [(120, 8), (122, 8), alto_six_musicmaker_two],  
1891 [(122, 8), (124, 8), alto_six_musicmaker_one],  
1892 [(124, 8), (126, 8), alto_six_musicmaker_one],  
1893 [(126, 8), (128, 8), alto_six_musicmaker_two],  
1894 [(128, 8), (130, 8), alto_six_musicmaker_three],  
1895 [(130, 8), (132, 8), alto_six_musicmaker_three],  
1896 [(132, 8), (134, 8), alto_six_musicmaker_three],  
1897 [(134, 8), (136, 8), alto_six_musicmaker_three],  
1898 [(136, 8), (138, 8), alto_six_musicmaker_three],  
1899 [(138, 8), (140, 8), alto_six_musicmaker_three],  
1900 [(140, 8), (142, 8), alto_six_musicmaker_one],  
1901 [(142, 8), (144, 8), alto_six_musicmaker_one],  
1902 [(144, 8), (146, 8), alto_six_musicmaker_two],  
1903 [(146, 8), (148, 8), alto_six_musicmaker_one],  
1904 [(148, 8), (150, 8), alto_six_musicmaker_two],  
1905 [(150, 8), (152, 8), alto_six_musicmaker_two],  
1906 [(152, 8), (154, 8), alto_six_musicmaker_three],  
1907 [(154, 8), (156, 8), alto_six_musicmaker_three],  
1908 [(156, 8), (158, 8), alto_six_musicmaker_three],  
1909 [(158, 8), (160, 8), alto_six_musicmaker_three],  
1910 [(160, 8), (162, 8), alto_six_musicmaker_three],  
1911 [(162, 8), (164, 8), alto_six_musicmaker_two],  
1912 [(164, 8), (166, 8), alto_six_musicmaker_one],  
1913 [(166, 8), (168, 8), alto_six_musicmaker_two],  
1914 [(168, 8), (170, 8), alto_six_musicmaker_one],  
1915 [(170, 8), (172, 8), alto_six_musicmaker_three],  
1916 [(172, 8), (174, 8), alto_six_musicmaker_three],  
1917 [(174, 8), (176, 8), alto_six_musicmaker_three],  
1918 [(176, 8), (178, 8), alto_six_musicmaker_three],  
1919 [(178, 8), (180, 8), alto_six_musicmaker_three],  
1920 [(180, 8), (182, 8), alto_six_musicmaker_one],  
1921 [(182, 8), (184, 8), alto_six_musicmaker_one],  
1922 [(184, 8), (186, 8), alto_six_musicmaker_one],  
1923 [(186, 8), (188, 8), alto_six_musicmaker_one],  
1924 [(188, 8), (190, 8), alto_six_musicmaker_three],  
1925 [(190, 8), (192, 8), alto_six_musicmaker_three],  
1926 [(192, 8), (194, 8), alto_six_musicmaker_three],  
1927 [(194, 8), (196, 8), alto_six_musicmaker_three],  
1928 [(196, 8), (198, 8), alto_six_musicmaker_three],  
1929 [(198, 8), (199, 8), alto_six_musicmaker_three],
```

```

1930     [(199, 8), (200, 8), alto_six_musicmaker_three],
1931   ]
1932 ])
1933
1934 voice_11_timespan_list = abjad.TimespanList([
1935   abjad.AnnotatedTimespan(
1936     start_offset=start_offset,
1937     stop_offset=stop_offset,
1938     annotation=MusicSpecifier(
1939       music_maker=music_maker,
1940       voice_name='Voice 11',
1941     ),
1942   )
1943   for start_offset, stop_offset, music_maker in [
1944     [(0, 8), (2, 8), tenor_one_musicmaker_three],
1945     [(2, 8), (4, 8), tenor_one_musicmaker_three],
1946     [(4, 8), (6, 8), tenor_one_musicmaker_one],
1947     [(6, 8), (8, 8), tenor_one_musicmaker_one],
1948     [(8, 8), (10, 8), tenor_one_musicmaker_one],
1949     [(10, 8), (12, 8), tenor_one_musicmaker_three],
1950     [(12, 8), (14, 8), tenor_one_musicmaker_three],
1951     [(14, 8), (16, 8), tenor_one_musicmaker_three],
1952     [(16, 8), (18, 8), tenor_one_musicmaker_three],
1953     [(18, 8), (20, 8), tenor_one_musicmaker_two],
1954     [(20, 8), (22, 8), tenor_one_musicmaker_one],
1955     [(22, 8), (24, 8), tenor_one_musicmaker_one],
1956     [(24, 8), (26, 8), tenor_one_musicmaker_one],
1957     [(26, 8), (28, 8), tenor_one_musicmaker_two],
1958     [(28, 8), (30, 8), tenor_one_musicmaker_three],
1959     [(30, 8), (32, 8), tenor_one_musicmaker_three],
1960     [(32, 8), (34, 8), tenor_one_musicmaker_three],
1961     [(34, 8), (36, 8), tenor_one_musicmaker_three],
1962     [(36, 8), (38, 8), tenor_one_musicmaker_two],
1963     [(38, 8), (40, 8), tenor_one_musicmaker_one],
1964     [(40, 8), (42, 8), tenor_one_musicmaker_two],
1965     [(42, 8), (44, 8), tenor_one_musicmaker_three],
1966     [(44, 8), (46, 8), tenor_one_musicmaker_three],
1967     [(46, 8), (48, 8), tenor_one_musicmaker_three],
1968     [(48, 8), (50, 8), tenor_one_musicmaker_three],
1969     [(50, 8), (52, 8), tenor_one_musicmaker_three],
1970     [(52, 8), (54, 8), tenor_one_musicmaker_two],
1971     [(54, 8), (56, 8), tenor_one_musicmaker_one],
1972     [(56, 8), (58, 8), tenor_one_musicmaker_one],
1973     [(58, 8), (60, 8), tenor_one_musicmaker_two],
1974     [(60, 8), (62, 8), tenor_one_musicmaker_three],
1975     [(62, 8), (64, 8), tenor_one_musicmaker_three],
1976     [(64, 8), (66, 8), tenor_one_musicmaker_one],
1977     [(66, 8), (68, 8), tenor_one_musicmaker_three],
1978     [(68, 8), (70, 8), tenor_one_musicmaker_three],
1979     [(70, 8), (72, 8), tenor_one_musicmaker_one],
1980     [(72, 8), (74, 8), tenor_one_musicmaker_one],
1981     [(74, 8), (76, 8), tenor_one_musicmaker_three],
1982     [(76, 8), (78, 8), tenor_one_musicmaker_three],
1983     [(78, 8), (80, 8), tenor_one_musicmaker_three],
1984     [(80, 8), (82, 8), tenor_one_musicmaker_three],
1985     [(82, 8), (84, 8), tenor_one_musicmaker_three],
1986     [(84, 8), (86, 8), tenor_one_musicmaker_three],
1987     [(86, 8), (88, 8), tenor_one_musicmaker_two],
1988     [(88, 8), (90, 8), tenor_one_musicmaker_one],

```

```

1989      [(90, 8), (92, 8), tenor_one_musicmaker_one],
1990      [(92, 8), (94, 8), tenor_one_musicmaker_two],
1991      [(94, 8), (96, 8), tenor_one_musicmaker_one],
1992      [(96, 8), (98, 8), tenor_one_musicmaker_three],
1993      [(98, 8), (100, 8), tenor_one_musicmaker_three],
1994      [(100, 8), (102, 8), tenor_one_musicmaker_three],
1995      [(102, 8), (104, 8), tenor_one_musicmaker_three],
1996      [(104, 8), (106, 8), tenor_one_musicmaker_three],
1997      [(106, 8), (108, 8), tenor_one_musicmaker_three],
1998      [(108, 8), (110, 8), tenor_one_musicmaker_three],
1999      [(110, 8), (112, 8), tenor_one_musicmaker_two],
2000      [(112, 8), (114, 8), tenor_one_musicmaker_one],
2001      [(114, 8), (116, 8), tenor_one_musicmaker_two],
2002      [(116, 8), (118, 8), tenor_one_musicmaker_one],
2003      [(118, 8), (120, 8), tenor_one_musicmaker_one],
2004      [(120, 8), (122, 8), tenor_one_musicmaker_two],
2005      [(122, 8), (124, 8), tenor_one_musicmaker_three],
2006      [(124, 8), (126, 8), tenor_one_musicmaker_three],
2007      [(126, 8), (128, 8), tenor_one_musicmaker_three],
2008      [(128, 8), (130, 8), tenor_one_musicmaker_three],
2009      [(130, 8), (132, 8), tenor_one_musicmaker_three],
2010      [(132, 8), (134, 8), tenor_one_musicmaker_three],
2011      [(134, 8), (136, 8), tenor_one_musicmaker_three],
2012      [(136, 8), (138, 8), tenor_one_musicmaker_three],
2013      [(138, 8), (140, 8), tenor_one_musicmaker_one],
2014      [(140, 8), (142, 8), tenor_one_musicmaker_two],
2015      [(142, 8), (144, 8), tenor_one_musicmaker_one],
2016      [(144, 8), (146, 8), tenor_one_musicmaker_one],
2017      [(146, 8), (148, 8), tenor_one_musicmaker_three],
2018      [(148, 8), (150, 8), tenor_one_musicmaker_three],
2019      [(150, 8), (152, 8), tenor_one_musicmaker_three],
2020      [(152, 8), (154, 8), tenor_one_musicmaker_three],
2021      [(154, 8), (156, 8), tenor_one_musicmaker_three],
2022      [(156, 8), (158, 8), tenor_one_musicmaker_two],
2023      [(158, 8), (160, 8), tenor_one_musicmaker_one],
2024      [(160, 8), (162, 8), tenor_one_musicmaker_two],
2025      [(162, 8), (164, 8), tenor_one_musicmaker_two],
2026      [(164, 8), (166, 8), tenor_one_musicmaker_three],
2027      [(166, 8), (168, 8), tenor_one_musicmaker_three],
2028      [(168, 8), (170, 8), tenor_one_musicmaker_three],
2029      [(170, 8), (172, 8), tenor_one_musicmaker_three],
2030      [(172, 8), (174, 8), tenor_one_musicmaker_two],
2031      [(174, 8), (176, 8), tenor_one_musicmaker_one],
2032      [(176, 8), (178, 8), tenor_one_musicmaker_two],
2033      [(178, 8), (180, 8), tenor_one_musicmaker_two],
2034      [(180, 8), (182, 8), tenor_one_musicmaker_one],
2035      [(182, 8), (184, 8), tenor_one_musicmaker_three],
2036      [(184, 8), (186, 8), tenor_one_musicmaker_three],
2037      [(186, 8), (188, 8), tenor_one_musicmaker_three],
2038      [(188, 8), (190, 8), tenor_one_musicmaker_three],
2039      [(190, 8), (192, 8), tenor_one_musicmaker_three],
2040      [(192, 8), (194, 8), tenor_one_musicmaker_three],
2041      [(194, 8), (196, 8), tenor_one_musicmaker_three],
2042      [(196, 8), (198, 8), tenor_one_musicmaker_two],
2043      [(198, 8), (199, 8), tenor_one_musicmaker_one],
2044      [(199, 8), (200, 8), tenor_one_musicmaker_three],
2045  ]
2046  ])
2047

```

```

2048 voice_12_timespan_list = abjad.TimespanList([
2049     abjad.AnnotatedTimespan(
2050         start_offset=start_offset,
2051         stop_offset=stop_offset,
2052         annotation=MusicSpecifier(
2053             music_maker=music_maker,
2054             voice_name='Voice 12',
2055         ),
2056     )
2057     for start_offset, stop_offset, music_maker in [
2058         [(0, 8), (2, 8), tenor_two_musicmaker_one],
2059         [(2, 8), (4, 8), tenor_two_musicmaker_two],
2060         [(4, 8), (6, 8), tenor_two_musicmaker_two],
2061         [(6, 8), (8, 8), tenor_two_musicmaker_three],
2062         [(8, 8), (10, 8), tenor_two_musicmaker_three],
2063         [(10, 8), (12, 8), tenor_two_musicmaker_three],
2064         [(12, 8), (14, 8), tenor_two_musicmaker_three],
2065         [(14, 8), (16, 8), tenor_two_musicmaker_one],
2066         [(16, 8), (18, 8), tenor_two_musicmaker_two],
2067         [(18, 8), (20, 8), tenor_two_musicmaker_one],
2068         [(20, 8), (22, 8), tenor_two_musicmaker_three],
2069         [(22, 8), (24, 8), tenor_two_musicmaker_three],
2070         [(24, 8), (26, 8), tenor_two_musicmaker_one],
2071         [(26, 8), (28, 8), tenor_two_musicmaker_two],
2072         [(28, 8), (30, 8), tenor_two_musicmaker_three],
2073         [(30, 8), (32, 8), tenor_two_musicmaker_three],
2074         [(32, 8), (34, 8), tenor_two_musicmaker_three],
2075         [(34, 8), (36, 8), tenor_two_musicmaker_three],
2076         [(36, 8), (38, 8), tenor_two_musicmaker_three],
2077         [(38, 8), (40, 8), tenor_two_musicmaker_three],
2078         [(40, 8), (42, 8), tenor_two_musicmaker_two],
2079         [(42, 8), (44, 8), tenor_two_musicmaker_one],
2080         [(44, 8), (46, 8), tenor_two_musicmaker_two],
2081         [(46, 8), (48, 8), tenor_two_musicmaker_one],
2082         [(48, 8), (50, 8), tenor_two_musicmaker_three],
2083         [(50, 8), (52, 8), tenor_two_musicmaker_three],
2084         [(52, 8), (54, 8), tenor_two_musicmaker_three],
2085         [(54, 8), (56, 8), tenor_two_musicmaker_three],
2086         [(56, 8), (58, 8), tenor_two_musicmaker_two],
2087         [(58, 8), (60, 8), tenor_two_musicmaker_one],
2088         [(60, 8), (62, 8), tenor_two_musicmaker_three],
2089         [(62, 8), (64, 8), tenor_two_musicmaker_three],
2090         [(64, 8), (66, 8), tenor_two_musicmaker_three],
2091         [(66, 8), (68, 8), tenor_two_musicmaker_three],
2092         [(68, 8), (70, 8), tenor_two_musicmaker_three],
2093         [(70, 8), (72, 8), tenor_two_musicmaker_three],
2094         [(72, 8), (74, 8), tenor_two_musicmaker_one],
2095         [(74, 8), (76, 8), tenor_two_musicmaker_two],
2096         [(76, 8), (78, 8), tenor_two_musicmaker_two],
2097         [(78, 8), (80, 8), tenor_two_musicmaker_one],
2098         [(80, 8), (82, 8), tenor_two_musicmaker_one],
2099         [(82, 8), (84, 8), tenor_two_musicmaker_three],
2100         [(84, 8), (86, 8), tenor_two_musicmaker_three],
2101         [(86, 8), (88, 8), tenor_two_musicmaker_three],
2102         [(88, 8), (90, 8), tenor_two_musicmaker_three],
2103         [(90, 8), (92, 8), tenor_two_musicmaker_three],
2104         [(92, 8), (94, 8), tenor_two_musicmaker_three],
2105         [(94, 8), (96, 8), tenor_two_musicmaker_three],
2106         [(96, 8), (98, 8), tenor_two_musicmaker_three],

```

```

2107     [(98, 8), (100, 8), tenor_two_musicmaker_three],
2108     [(100, 8), (102, 8), tenor_two_musicmaker_two],
2109     [(102, 8), (104, 8), tenor_two_musicmaker_one],
2110     [(104, 8), (106, 8), tenor_two_musicmaker_one],
2111     [(106, 8), (108, 8), tenor_two_musicmaker_three],
2112     [(108, 8), (110, 8), tenor_two_musicmaker_three],
2113     [(110, 8), (112, 8), tenor_two_musicmaker_three],
2114     [(112, 8), (114, 8), tenor_two_musicmaker_three],
2115     [(114, 8), (116, 8), tenor_two_musicmaker_one],
2116     [(116, 8), (118, 8), tenor_two_musicmaker_two],
2117     [(118, 8), (120, 8), tenor_two_musicmaker_two],
2118     [(120, 8), (122, 8), tenor_two_musicmaker_three],
2119     [(122, 8), (124, 8), tenor_two_musicmaker_three],
2120     [(124, 8), (126, 8), tenor_two_musicmaker_three],
2121     [(126, 8), (128, 8), tenor_two_musicmaker_three],
2122     [(128, 8), (130, 8), tenor_two_musicmaker_three],
2123     [(130, 8), (132, 8), tenor_two_musicmaker_three],
2124     [(132, 8), (134, 8), tenor_two_musicmaker_two],
2125     [(134, 8), (136, 8), tenor_two_musicmaker_one],
2126     [(136, 8), (138, 8), tenor_two_musicmaker_two],
2127     [(138, 8), (140, 8), tenor_two_musicmaker_one],
2128     [(140, 8), (142, 8), tenor_two_musicmaker_one],
2129     [(142, 8), (144, 8), tenor_two_musicmaker_three],
2130     [(144, 8), (146, 8), tenor_two_musicmaker_three],
2131     [(146, 8), (148, 8), tenor_two_musicmaker_three],
2132     [(148, 8), (150, 8), tenor_two_musicmaker_three],
2133     [(150, 8), (152, 8), tenor_two_musicmaker_three],
2134     [(152, 8), (154, 8), tenor_two_musicmaker_three],
2135     [(154, 8), (156, 8), tenor_two_musicmaker_two],
2136     [(156, 8), (158, 8), tenor_two_musicmaker_one],
2137     [(158, 8), (160, 8), tenor_two_musicmaker_two],
2138     [(160, 8), (162, 8), tenor_two_musicmaker_two],
2139     [(162, 8), (164, 8), tenor_two_musicmaker_three],
2140     [(164, 8), (166, 8), tenor_two_musicmaker_three],
2141     [(166, 8), (168, 8), tenor_two_musicmaker_three],
2142     [(168, 8), (170, 8), tenor_two_musicmaker_three],
2143     [(170, 8), (172, 8), tenor_two_musicmaker_three],
2144     [(172, 8), (174, 8), tenor_two_musicmaker_two],
2145     [(174, 8), (176, 8), tenor_two_musicmaker_one],
2146     [(176, 8), (178, 8), tenor_two_musicmaker_one],
2147     [(178, 8), (180, 8), tenor_two_musicmaker_one],
2148     [(180, 8), (182, 8), tenor_two_musicmaker_three],
2149     [(182, 8), (184, 8), tenor_two_musicmaker_three],
2150     [(184, 8), (186, 8), tenor_two_musicmaker_three],
2151     [(186, 8), (188, 8), tenor_two_musicmaker_two],
2152     [(188, 8), (190, 8), tenor_two_musicmaker_one],
2153     [(190, 8), (192, 8), tenor_two_musicmaker_two],
2154     [(192, 8), (194, 8), tenor_two_musicmaker_two],
2155     [(194, 8), (196, 8), tenor_two_musicmaker_three],
2156     [(196, 8), (198, 8), tenor_two_musicmaker_three],
2157     [(198, 8), (199, 8), tenor_two_musicmaker_two],
2158     [(199, 8), (200, 8), tenor_two_musicmaker_two],
2159   ]
2160 ])
2161
2162 voice_13_timespan_list = abjad.TimespanList([
2163     abjad.AnnotatedTimespan(
2164         start_offset=start_offset,
2165         stop_offset=stop_offset,

```

```
2166     annotation=MusicSpecifier(
2167         music_maker=music_maker,
2168         voice_name='Voice 13',
2169     ),
2170 )
2171 for start_offset, stop_offset, music_maker in [
2172     [(0, 8), (2, 8), tenor_three_musicmaker_three],
2173     [(2, 8), (4, 8), tenor_three_musicmaker_three],
2174     [(4, 8), (6, 8), tenor_three_musicmaker_three],
2175     [(6, 8), (8, 8), tenor_three_musicmaker_three],
2176     [(8, 8), (10, 8), tenor_three_musicmaker_three],
2177     [(10, 8), (12, 8), tenor_three_musicmaker_three],
2178     [(12, 8), (14, 8), tenor_three_musicmaker_one],
2179     [(14, 8), (16, 8), tenor_three_musicmaker_one],
2180     [(16, 8), (18, 8), tenor_three_musicmaker_two],
2181     [(18, 8), (20, 8), tenor_three_musicmaker_one],
2182     [(20, 8), (22, 8), tenor_three_musicmaker_one],
2183     [(22, 8), (24, 8), tenor_three_musicmaker_one],
2184     [(24, 8), (26, 8), tenor_three_musicmaker_two],
2185     [(26, 8), (28, 8), tenor_three_musicmaker_three],
2186     [(28, 8), (30, 8), tenor_three_musicmaker_three],
2187     [(30, 8), (32, 8), tenor_three_musicmaker_one],
2188     [(32, 8), (34, 8), tenor_three_musicmaker_two],
2189     [(34, 8), (36, 8), tenor_three_musicmaker_one],
2190     [(36, 8), (38, 8), tenor_three_musicmaker_one],
2191     [(38, 8), (40, 8), tenor_three_musicmaker_three],
2192     [(40, 8), (42, 8), tenor_three_musicmaker_three],
2193     [(42, 8), (44, 8), tenor_three_musicmaker_three],
2194     [(44, 8), (46, 8), tenor_three_musicmaker_three],
2195     [(46, 8), (48, 8), tenor_three_musicmaker_three],
2196     [(48, 8), (50, 8), tenor_three_musicmaker_three],
2197     [(50, 8), (52, 8), tenor_three_musicmaker_three],
2198     [(52, 8), (54, 8), tenor_three_musicmaker_three],
2199     [(54, 8), (56, 8), tenor_three_musicmaker_one],
2200     [(56, 8), (58, 8), tenor_three_musicmaker_two],
2201     [(58, 8), (60, 8), tenor_three_musicmaker_two],
2202     [(60, 8), (62, 8), tenor_three_musicmaker_one],
2203     [(62, 8), (64, 8), tenor_three_musicmaker_two],
2204     [(64, 8), (66, 8), tenor_three_musicmaker_three],
2205     [(66, 8), (68, 8), tenor_three_musicmaker_three],
2206     [(68, 8), (70, 8), tenor_three_musicmaker_three],
2207     [(70, 8), (72, 8), tenor_three_musicmaker_three],
2208     [(72, 8), (74, 8), tenor_three_musicmaker_one],
2209     [(74, 8), (76, 8), tenor_three_musicmaker_two],
2210     [(76, 8), (78, 8), tenor_three_musicmaker_one],
2211     [(78, 8), (80, 8), tenor_three_musicmaker_one],
2212     [(80, 8), (82, 8), tenor_three_musicmaker_two],
2213     [(82, 8), (84, 8), tenor_three_musicmaker_one],
2214     [(84, 8), (86, 8), tenor_three_musicmaker_one],
2215     [(86, 8), (88, 8), tenor_three_musicmaker_three],
2216     [(88, 8), (90, 8), tenor_three_musicmaker_three],
2217     [(90, 8), (92, 8), tenor_three_musicmaker_three],
2218     [(92, 8), (94, 8), tenor_three_musicmaker_three],
2219     [(94, 8), (96, 8), tenor_three_musicmaker_three],
2220     [(96, 8), (98, 8), tenor_three_musicmaker_three],
2221     [(98, 8), (100, 8), tenor_three_musicmaker_one],
2222     [(100, 8), (102, 8), tenor_three_musicmaker_one],
2223     [(102, 8), (104, 8), tenor_three_musicmaker_one],
2224     [(104, 8), (106, 8), tenor_three_musicmaker_three],
```

```

2225     [(106, 8), (108, 8), tenor_three_musicmaker_three],
2226     [(108, 8), (110, 8), tenor_three_musicmaker_three],
2227     [(110, 8), (112, 8), tenor_three_musicmaker_three],
2228     [(112, 8), (114, 8), tenor_three_musicmaker_three],
2229     [(114, 8), (116, 8), tenor_three_musicmaker_two],
2230     [(116, 8), (118, 8), tenor_three_musicmaker_one],
2231     [(118, 8), (120, 8), tenor_three_musicmaker_two],
2232     [(120, 8), (122, 8), tenor_three_musicmaker_one],
2233     [(122, 8), (124, 8), tenor_three_musicmaker_two],
2234     [(124, 8), (126, 8), tenor_three_musicmaker_two],
2235     [(126, 8), (128, 8), tenor_three_musicmaker_three],
2236     [(128, 8), (130, 8), tenor_three_musicmaker_three],
2237     [(130, 8), (132, 8), tenor_three_musicmaker_three],
2238     [(132, 8), (134, 8), tenor_three_musicmaker_three],
2239     [(134, 8), (136, 8), tenor_three_musicmaker_three],
2240     [(136, 8), (138, 8), tenor_three_musicmaker_three],
2241     [(138, 8), (140, 8), tenor_three_musicmaker_one],
2242     [(140, 8), (142, 8), tenor_three_musicmaker_three],
2243     [(142, 8), (144, 8), tenor_three_musicmaker_three],
2244     [(144, 8), (146, 8), tenor_three_musicmaker_one],
2245     [(146, 8), (148, 8), tenor_three_musicmaker_three],
2246     [(148, 8), (150, 8), tenor_three_musicmaker_three],
2247     [(150, 8), (152, 8), tenor_three_musicmaker_one],
2248     [(152, 8), (154, 8), tenor_three_musicmaker_one],
2249     [(154, 8), (156, 8), tenor_three_musicmaker_three],
2250     [(156, 8), (158, 8), tenor_three_musicmaker_three],
2251     [(158, 8), (160, 8), tenor_three_musicmaker_three],
2252     [(160, 8), (162, 8), tenor_three_musicmaker_one],
2253     [(162, 8), (164, 8), tenor_three_musicmaker_two],
2254     [(164, 8), (166, 8), tenor_three_musicmaker_two],
2255     [(166, 8), (168, 8), tenor_three_musicmaker_three],
2256     [(168, 8), (170, 8), tenor_three_musicmaker_three],
2257     [(170, 8), (172, 8), tenor_three_musicmaker_three],
2258     [(172, 8), (174, 8), tenor_three_musicmaker_three],
2259     [(174, 8), (176, 8), tenor_three_musicmaker_three],
2260     [(176, 8), (178, 8), tenor_three_musicmaker_three],
2261     [(178, 8), (180, 8), tenor_three_musicmaker_three],
2262     [(180, 8), (182, 8), tenor_three_musicmaker_three],
2263     [(182, 8), (184, 8), tenor_three_musicmaker_three],
2264     [(184, 8), (186, 8), tenor_three_musicmaker_two],
2265     [(186, 8), (188, 8), tenor_three_musicmaker_two],
2266     [(188, 8), (190, 8), tenor_three_musicmaker_two],
2267     [(190, 8), (192, 8), tenor_three_musicmaker_two],
2268     [(192, 8), (194, 8), tenor_three_musicmaker_two],
2269     [(194, 8), (196, 8), tenor_three_musicmaker_one],
2270     [(196, 8), (198, 8), tenor_three_musicmaker_three],
2271     [(198, 8), (199, 8), tenor_three_musicmaker_three],
2272     [(199, 8), (200, 8), tenor_three_musicmaker_three],
2273   ]
2274 ])
2275
2276 voice_14_timespan_list = abjad.TimespanList([
2277     abjad.AnnotatedTimespan(
2278         start_offset=start_offset,
2279         stop_offset=stop_offset,
2280         annotation=MusicSpecifier(
2281             music_maker=music_maker,
2282             voice_name='Voice 14',
2283         ),

```

```

2284 )
2285 for start_offset, stop_offset, music_maker in [
2286     [(0, 8), (2, 8), tenor_four_musicmaker_one],
2287     [(2, 8), (4, 8), tenor_four_musicmaker_one],
2288     [(4, 8), (6, 8), tenor_four_musicmaker_two],
2289     [(6, 8), (8, 8), tenor_four_musicmaker_one],
2290     [(8, 8), (10, 8), tenor_four_musicmaker_one],
2291     [(10, 8), (12, 8), tenor_four_musicmaker_one],
2292     [(12, 8), (14, 8), tenor_four_musicmaker_one],
2293     [(14, 8), (16, 8), tenor_four_musicmaker_three],
2294     [(16, 8), (18, 8), tenor_four_musicmaker_three],
2295     [(18, 8), (20, 8), tenor_four_musicmaker_three],
2296     [(20, 8), (22, 8), tenor_four_musicmaker_three],
2297     [(22, 8), (24, 8), tenor_four_musicmaker_three],
2298     [(24, 8), (26, 8), tenor_four_musicmaker_three],
2299     [(26, 8), (28, 8), tenor_four_musicmaker_three],
2300     [(28, 8), (30, 8), tenor_four_musicmaker_three],
2301     [(30, 8), (32, 8), tenor_four_musicmaker_three],
2302     [(32, 8), (34, 8), tenor_four_musicmaker_three],
2303     [(34, 8), (36, 8), tenor_four_musicmaker_two],
2304     [(36, 8), (38, 8), tenor_four_musicmaker_one],
2305     [(38, 8), (40, 8), tenor_four_musicmaker_two],
2306     [(40, 8), (42, 8), tenor_four_musicmaker_one],
2307     [(42, 8), (44, 8), tenor_four_musicmaker_one],
2308     [(44, 8), (46, 8), tenor_four_musicmaker_one],
2309     [(46, 8), (48, 8), tenor_four_musicmaker_one],
2310     [(48, 8), (50, 8), tenor_four_musicmaker_three],
2311     [(50, 8), (52, 8), tenor_four_musicmaker_three],
2312     [(52, 8), (54, 8), tenor_four_musicmaker_three],
2313     [(54, 8), (56, 8), tenor_four_musicmaker_three],
2314     [(56, 8), (58, 8), tenor_four_musicmaker_three],
2315     [(58, 8), (60, 8), tenor_four_musicmaker_three],
2316     [(60, 8), (62, 8), tenor_four_musicmaker_two],
2317     [(62, 8), (64, 8), tenor_four_musicmaker_two],
2318     [(64, 8), (66, 8), tenor_four_musicmaker_one],
2319     [(66, 8), (68, 8), tenor_four_musicmaker_two],
2320     [(68, 8), (70, 8), tenor_four_musicmaker_one],
2321     [(70, 8), (72, 8), tenor_four_musicmaker_one],
2322     [(72, 8), (74, 8), tenor_four_musicmaker_three],
2323     [(74, 8), (76, 8), tenor_four_musicmaker_three],
2324     [(76, 8), (78, 8), tenor_four_musicmaker_three],
2325     [(78, 8), (80, 8), tenor_four_musicmaker_three],
2326     [(80, 8), (82, 8), tenor_four_musicmaker_three],
2327     [(82, 8), (84, 8), tenor_four_musicmaker_three],
2328     [(84, 8), (86, 8), tenor_four_musicmaker_three],
2329     [(86, 8), (88, 8), tenor_four_musicmaker_one],
2330     [(88, 8), (90, 8), tenor_four_musicmaker_two],
2331     [(90, 8), (92, 8), tenor_four_musicmaker_one],
2332     [(92, 8), (94, 8), tenor_four_musicmaker_two],
2333     [(94, 8), (96, 8), tenor_four_musicmaker_three],
2334     [(96, 8), (98, 8), tenor_four_musicmaker_three],
2335     [(98, 8), (100, 8), tenor_four_musicmaker_three],
2336     [(100, 8), (102, 8), tenor_four_musicmaker_one],
2337     [(102, 8), (104, 8), tenor_four_musicmaker_three],
2338     [(104, 8), (106, 8), tenor_four_musicmaker_three],
2339     [(106, 8), (108, 8), tenor_four_musicmaker_three],
2340     [(108, 8), (110, 8), tenor_four_musicmaker_two],
2341     [(110, 8), (112, 8), tenor_four_musicmaker_one],
2342     [(112, 8), (114, 8), tenor_four_musicmaker_one],

```

```

2343     [(114, 8), (116, 8), tenor_four_musicmaker_two],
2344     [(116, 8), (118, 8), tenor_four_musicmaker_two],
2345     [(118, 8), (120, 8), tenor_four_musicmaker_three],
2346     [(120, 8), (122, 8), tenor_four_musicmaker_three],
2347     [(122, 8), (124, 8), tenor_four_musicmaker_three],
2348     [(124, 8), (126, 8), tenor_four_musicmaker_three],
2349     [(126, 8), (128, 8), tenor_four_musicmaker_one],
2350     [(128, 8), (130, 8), tenor_four_musicmaker_two],
2351     [(130, 8), (132, 8), tenor_four_musicmaker_one],
2352     [(132, 8), (134, 8), tenor_four_musicmaker_one],
2353     [(134, 8), (136, 8), tenor_four_musicmaker_two],
2354     [(136, 8), (138, 8), tenor_four_musicmaker_three],
2355     [(138, 8), (140, 8), tenor_four_musicmaker_three],
2356     [(140, 8), (142, 8), tenor_four_musicmaker_three],
2357     [(142, 8), (144, 8), tenor_four_musicmaker_three],
2358     [(144, 8), (146, 8), tenor_four_musicmaker_three],
2359     [(146, 8), (148, 8), tenor_four_musicmaker_two],
2360     [(148, 8), (150, 8), tenor_four_musicmaker_one],
2361     [(150, 8), (152, 8), tenor_four_musicmaker_one],
2362     [(152, 8), (154, 8), tenor_four_musicmaker_two],
2363     [(154, 8), (156, 8), tenor_four_musicmaker_one],
2364     [(156, 8), (158, 8), tenor_four_musicmaker_one],
2365     [(158, 8), (160, 8), tenor_four_musicmaker_two],
2366     [(160, 8), (162, 8), tenor_four_musicmaker_three],
2367     [(162, 8), (164, 8), tenor_four_musicmaker_three],
2368     [(164, 8), (166, 8), tenor_four_musicmaker_three],
2369     [(166, 8), (168, 8), tenor_four_musicmaker_three],
2370     [(168, 8), (170, 8), tenor_four_musicmaker_three],
2371     [(170, 8), (172, 8), tenor_four_musicmaker_three],
2372     [(172, 8), (174, 8), tenor_four_musicmaker_three],
2373     [(174, 8), (176, 8), tenor_four_musicmaker_three],
2374     [(176, 8), (178, 8), tenor_four_musicmaker_three],
2375     [(178, 8), (180, 8), tenor_four_musicmaker_two],
2376     [(180, 8), (182, 8), tenor_four_musicmaker_two],
2377     [(182, 8), (184, 8), tenor_four_musicmaker_two],
2378     [(184, 8), (186, 8), tenor_four_musicmaker_one],
2379     [(186, 8), (188, 8), tenor_four_musicmaker_two],
2380     [(188, 8), (190, 8), tenor_four_musicmaker_one],
2381     [(190, 8), (192, 8), tenor_four_musicmaker_two],
2382     [(192, 8), (194, 8), tenor_four_musicmaker_two],
2383     [(194, 8), (196, 8), tenor_four_musicmaker_two],
2384     [(196, 8), (198, 8), tenor_four_musicmaker_one],
2385     [(198, 8), (199, 8), tenor_four_musicmaker_three],
2386     [(199, 8), (200, 8), tenor_four_musicmaker_three],
2387   ],
2388 ])
2389
2390 voice_15_timespan_list = abjad.TimespanList([
2391     abjad.AnnotatedTimespan(
2392         start_offset=start_offset,
2393         stop_offset=stop_offset,
2394         annotation=MusicSpecifier(
2395             music_maker=music_maker,
2396             voice_name='Voice 15',
2397         ),
2398     )
2399     for start_offset, stop_offset, music_maker in [
2400         [(0, 8), (2, 8), tenor_five_musicmaker_three],
2401         [(2, 8), (4, 8), tenor_five_musicmaker_one],

```

```
2402      [(4, 8), (6, 8), tenor_five_musicmaker_three],  
2403      [(6, 8), (8, 8), tenor_five_musicmaker_three],  
2404      [(8, 8), (10, 8), tenor_five_musicmaker_three],  
2405      [(10, 8), (12, 8), tenor_five_musicmaker_three],  
2406      [(12, 8), (14, 8), tenor_five_musicmaker_two],  
2407      [(14, 8), (16, 8), tenor_five_musicmaker_one],  
2408      [(16, 8), (18, 8), tenor_five_musicmaker_one],  
2409      [(18, 8), (20, 8), tenor_five_musicmaker_two],  
2410      [(20, 8), (22, 8), tenor_five_musicmaker_two],  
2411      [(22, 8), (24, 8), tenor_five_musicmaker_three],  
2412      [(24, 8), (26, 8), tenor_five_musicmaker_three],  
2413      [(26, 8), (28, 8), tenor_five_musicmaker_three],  
2414      [(28, 8), (30, 8), tenor_five_musicmaker_three],  
2415      [(30, 8), (32, 8), tenor_five_musicmaker_three],  
2416      [(32, 8), (34, 8), tenor_five_musicmaker_one],  
2417      [(34, 8), (36, 8), tenor_five_musicmaker_three],  
2418      [(36, 8), (38, 8), tenor_five_musicmaker_three],  
2419      [(38, 8), (40, 8), tenor_five_musicmaker_three],  
2420      [(40, 8), (42, 8), tenor_five_musicmaker_three],  
2421      [(42, 8), (44, 8), tenor_five_musicmaker_one],  
2422      [(44, 8), (46, 8), tenor_five_musicmaker_two],  
2423      [(46, 8), (48, 8), tenor_five_musicmaker_two],  
2424      [(48, 8), (50, 8), tenor_five_musicmaker_two],  
2425      [(50, 8), (52, 8), tenor_five_musicmaker_two],  
2426      [(52, 8), (54, 8), tenor_five_musicmaker_three],  
2427      [(54, 8), (56, 8), tenor_five_musicmaker_three],  
2428      [(56, 8), (58, 8), tenor_five_musicmaker_three],  
2429      [(58, 8), (60, 8), tenor_five_musicmaker_three],  
2430      [(60, 8), (62, 8), tenor_five_musicmaker_three],  
2431      [(62, 8), (64, 8), tenor_five_musicmaker_three],  
2432      [(64, 8), (66, 8), tenor_five_musicmaker_three],  
2433      [(66, 8), (68, 8), tenor_five_musicmaker_three],  
2434      [(68, 8), (70, 8), tenor_five_musicmaker_three],  
2435      [(70, 8), (72, 8), tenor_five_musicmaker_three],  
2436      [(72, 8), (74, 8), tenor_five_musicmaker_two],  
2437      [(74, 8), (76, 8), tenor_five_musicmaker_one],  
2438      [(76, 8), (78, 8), tenor_five_musicmaker_one],  
2439      [(78, 8), (80, 8), tenor_five_musicmaker_one],  
2440      [(80, 8), (82, 8), tenor_five_musicmaker_one],  
2441      [(82, 8), (84, 8), tenor_five_musicmaker_one],  
2442      [(84, 8), (86, 8), tenor_five_musicmaker_three],  
2443      [(86, 8), (88, 8), tenor_five_musicmaker_three],  
2444      [(88, 8), (90, 8), tenor_five_musicmaker_one],  
2445      [(90, 8), (92, 8), tenor_five_musicmaker_two],  
2446      [(92, 8), (94, 8), tenor_five_musicmaker_two],  
2447      [(94, 8), (96, 8), tenor_five_musicmaker_three],  
2448      [(96, 8), (98, 8), tenor_five_musicmaker_three],  
2449      [(98, 8), (100, 8), tenor_five_musicmaker_three],  
2450      [(100, 8), (102, 8), tenor_five_musicmaker_three],  
2451      [(102, 8), (104, 8), tenor_five_musicmaker_three],  
2452      [(104, 8), (106, 8), tenor_five_musicmaker_three],  
2453      [(106, 8), (108, 8), tenor_five_musicmaker_three],  
2454      [(108, 8), (110, 8), tenor_five_musicmaker_two],  
2455      [(110, 8), (112, 8), tenor_five_musicmaker_one],  
2456      [(112, 8), (114, 8), tenor_five_musicmaker_three],  
2457      [(114, 8), (116, 8), tenor_five_musicmaker_three],  
2458      [(116, 8), (118, 8), tenor_five_musicmaker_one],  
2459      [(118, 8), (120, 8), tenor_five_musicmaker_two],  
2460      [(120, 8), (122, 8), tenor_five_musicmaker_two],
```

```

2461     [(122, 8), (124, 8), tenor_five_musicmaker_three],
2462     [(124, 8), (126, 8), tenor_five_musicmaker_three],
2463     [(126, 8), (128, 8), tenor_five_musicmaker_three],
2464     [(128, 8), (130, 8), tenor_five_musicmaker_three],
2465     [(130, 8), (132, 8), tenor_five_musicmaker_three],
2466     [(132, 8), (134, 8), tenor_five_musicmaker_two],
2467     [(134, 8), (136, 8), tenor_five_musicmaker_one],
2468     [(136, 8), (138, 8), tenor_five_musicmaker_one],
2469     [(138, 8), (140, 8), tenor_five_musicmaker_three],
2470     [(140, 8), (142, 8), tenor_five_musicmaker_three],
2471     [(142, 8), (144, 8), tenor_five_musicmaker_one],
2472     [(144, 8), (146, 8), tenor_five_musicmaker_two],
2473     [(146, 8), (148, 8), tenor_five_musicmaker_one],
2474     [(148, 8), (150, 8), tenor_five_musicmaker_two],
2475     [(150, 8), (152, 8), tenor_five_musicmaker_one],
2476     [(152, 8), (154, 8), tenor_five_musicmaker_one],
2477     [(154, 8), (156, 8), tenor_five_musicmaker_three],
2478     [(156, 8), (158, 8), tenor_five_musicmaker_three],
2479     [(158, 8), (160, 8), tenor_five_musicmaker_three],
2480     [(160, 8), (162, 8), tenor_five_musicmaker_three],
2481     [(162, 8), (164, 8), tenor_five_musicmaker_one],
2482     [(164, 8), (166, 8), tenor_five_musicmaker_two],
2483     [(166, 8), (168, 8), tenor_five_musicmaker_three],
2484     [(168, 8), (170, 8), tenor_five_musicmaker_three],
2485     [(170, 8), (172, 8), tenor_five_musicmaker_three],
2486     [(172, 8), (174, 8), tenor_five_musicmaker_one],
2487     [(174, 8), (176, 8), tenor_five_musicmaker_one],
2488     [(176, 8), (178, 8), tenor_five_musicmaker_three],
2489     [(178, 8), (180, 8), tenor_five_musicmaker_three],
2490     [(180, 8), (182, 8), tenor_five_musicmaker_three],
2491     [(182, 8), (184, 8), tenor_five_musicmaker_three],
2492     [(184, 8), (186, 8), tenor_five_musicmaker_one],
2493     [(186, 8), (188, 8), tenor_five_musicmaker_one],
2494     [(188, 8), (190, 8), tenor_five_musicmaker_one],
2495     [(190, 8), (192, 8), tenor_five_musicmaker_two],
2496     [(192, 8), (194, 8), tenor_five_musicmaker_three],
2497     [(194, 8), (196, 8), tenor_five_musicmaker_three],
2498     [(196, 8), (198, 8), tenor_five_musicmaker_three],
2499     [(198, 8), (199, 8), tenor_five_musicmaker_three],
2500     [(199, 8), (200, 8), tenor_five_musicmaker_three],
2501 ]
2502 ])
2503
2504 voice_16_timespan_list = abjad.TimespanList([
2505     abjad.AnnotatedTimespan(
2506         start_offset=start_offset,
2507         stop_offset=stop_offset,
2508         annotation=MusicSpecifier(
2509             music_maker=music_maker,
2510             voice_name='Voice 16',
2511         ),
2512     )
2513     for start_offset, stop_offset, music_maker in [
2514         [(0, 8), (2, 8), baritone_one_musicmaker_one],
2515         [(2, 8), (4, 8), baritone_one_musicmaker_one],
2516         [(4, 8), (6, 8), baritone_one_musicmaker_two],
2517         [(6, 8), (8, 8), baritone_one_musicmaker_one],
2518         [(8, 8), (10, 8), baritone_one_musicmaker_three],
2519         [(10, 8), (12, 8), baritone_one_musicmaker_three],

```

```
2520      [(12, 8), (14, 8), baritone_one_musicmaker_three],  
2521      [(14, 8), (16, 8), baritone_one_musicmaker_one],  
2522      [(16, 8), (18, 8), baritone_one_musicmaker_two],  
2523      [(18, 8), (20, 8), baritone_one_musicmaker_one],  
2524      [(20, 8), (22, 8), baritone_one_musicmaker_one],  
2525      [(22, 8), (24, 8), baritone_one_musicmaker_two],  
2526      [(24, 8), (26, 8), baritone_one_musicmaker_three],  
2527      [(26, 8), (28, 8), baritone_one_musicmaker_three],  
2528      [(28, 8), (30, 8), baritone_one_musicmaker_three],  
2529      [(30, 8), (32, 8), baritone_one_musicmaker_three],  
2530      [(32, 8), (34, 8), baritone_one_musicmaker_one],  
2531      [(34, 8), (36, 8), baritone_one_musicmaker_two],  
2532      [(36, 8), (38, 8), baritone_one_musicmaker_two],  
2533      [(38, 8), (40, 8), baritone_one_musicmaker_three],  
2534      [(40, 8), (42, 8), baritone_one_musicmaker_three],  
2535      [(42, 8), (44, 8), baritone_one_musicmaker_three],  
2536      [(44, 8), (46, 8), baritone_one_musicmaker_one],  
2537      [(46, 8), (48, 8), baritone_one_musicmaker_one],  
2538      [(48, 8), (50, 8), baritone_one_musicmaker_one],  
2539      [(50, 8), (52, 8), baritone_one_musicmaker_three],  
2540      [(52, 8), (54, 8), baritone_one_musicmaker_three],  
2541      [(54, 8), (56, 8), baritone_one_musicmaker_three],  
2542      [(56, 8), (58, 8), baritone_one_musicmaker_one],  
2543      [(58, 8), (60, 8), baritone_one_musicmaker_two],  
2544      [(60, 8), (62, 8), baritone_one_musicmaker_one],  
2545      [(62, 8), (64, 8), baritone_one_musicmaker_two],  
2546      [(64, 8), (66, 8), baritone_one_musicmaker_two],  
2547      [(66, 8), (68, 8), baritone_one_musicmaker_three],  
2548      [(68, 8), (70, 8), baritone_one_musicmaker_three],  
2549      [(70, 8), (72, 8), baritone_one_musicmaker_three],  
2550      [(72, 8), (74, 8), baritone_one_musicmaker_one],  
2551      [(74, 8), (76, 8), baritone_one_musicmaker_one],  
2552      [(76, 8), (78, 8), baritone_one_musicmaker_one],  
2553      [(78, 8), (80, 8), baritone_one_musicmaker_three],  
2554      [(80, 8), (82, 8), baritone_one_musicmaker_three],  
2555      [(82, 8), (84, 8), baritone_one_musicmaker_three],  
2556      [(84, 8), (86, 8), baritone_one_musicmaker_one],  
2557      [(86, 8), (88, 8), baritone_one_musicmaker_two],  
2558      [(88, 8), (90, 8), baritone_one_musicmaker_one],  
2559      [(90, 8), (92, 8), baritone_one_musicmaker_three],  
2560      [(92, 8), (94, 8), baritone_one_musicmaker_three],  
2561      [(94, 8), (96, 8), baritone_one_musicmaker_three],  
2562      [(96, 8), (98, 8), baritone_one_musicmaker_three],  
2563      [(98, 8), (100, 8), baritone_one_musicmaker_three],  
2564      [(100, 8), (102, 8), baritone_one_musicmaker_one],  
2565      [(102, 8), (104, 8), baritone_one_musicmaker_one],  
2566      [(104, 8), (106, 8), baritone_one_musicmaker_two],  
2567      [(106, 8), (108, 8), baritone_one_musicmaker_three],  
2568      [(108, 8), (110, 8), baritone_one_musicmaker_three],  
2569      [(110, 8), (112, 8), baritone_one_musicmaker_three],  
2570      [(112, 8), (114, 8), baritone_one_musicmaker_three],  
2571      [(114, 8), (116, 8), baritone_one_musicmaker_one],  
2572      [(116, 8), (118, 8), baritone_one_musicmaker_three],  
2573      [(118, 8), (120, 8), baritone_one_musicmaker_three],  
2574      [(120, 8), (122, 8), baritone_one_musicmaker_three],  
2575      [(122, 8), (124, 8), baritone_one_musicmaker_three],  
2576      [(124, 8), (126, 8), baritone_one_musicmaker_three],  
2577      [(126, 8), (128, 8), baritone_one_musicmaker_two],  
2578      [(128, 8), (130, 8), baritone_one_musicmaker_two],
```

```

2579     [(130, 8), (132, 8), baritone_one_musicmaker_two],
2580     [(132, 8), (134, 8), baritone_one_musicmaker_one],
2581     [(134, 8), (136, 8), baritone_one_musicmaker_two],
2582     [(136, 8), (138, 8), baritone_one_musicmaker_three],
2583     [(138, 8), (140, 8), baritone_one_musicmaker_three],
2584     [(140, 8), (142, 8), baritone_one_musicmaker_three],
2585     [(142, 8), (144, 8), baritone_one_musicmaker_three],
2586     [(144, 8), (146, 8), baritone_one_musicmaker_one],
2587     [(146, 8), (148, 8), baritone_one_musicmaker_one],
2588     [(148, 8), (150, 8), baritone_one_musicmaker_three],
2589     [(150, 8), (152, 8), baritone_one_musicmaker_three],
2590     [(152, 8), (154, 8), baritone_one_musicmaker_three],
2591     [(154, 8), (156, 8), baritone_one_musicmaker_three],
2592     [(156, 8), (158, 8), baritone_one_musicmaker_three],
2593     [(158, 8), (160, 8), baritone_one_musicmaker_three],
2594     [(160, 8), (162, 8), baritone_one_musicmaker_one],
2595     [(162, 8), (164, 8), baritone_one_musicmaker_two],
2596     [(164, 8), (166, 8), baritone_one_musicmaker_three],
2597     [(166, 8), (168, 8), baritone_one_musicmaker_three],
2598     [(168, 8), (170, 8), baritone_one_musicmaker_three],
2599     [(170, 8), (172, 8), baritone_one_musicmaker_one],
2600     [(172, 8), (174, 8), baritone_one_musicmaker_two],
2601     [(174, 8), (176, 8), baritone_one_musicmaker_two],
2602     [(176, 8), (178, 8), baritone_one_musicmaker_three],
2603     [(178, 8), (180, 8), baritone_one_musicmaker_three],
2604     [(180, 8), (182, 8), baritone_one_musicmaker_three],
2605     [(182, 8), (184, 8), baritone_one_musicmaker_three],
2606     [(184, 8), (186, 8), baritone_one_musicmaker_two],
2607     [(186, 8), (188, 8), baritone_one_musicmaker_one],
2608     [(188, 8), (190, 8), baritone_one_musicmaker_one],
2609     [(190, 8), (192, 8), baritone_one_musicmaker_two],
2610     [(192, 8), (194, 8), baritone_one_musicmaker_three],
2611     [(194, 8), (196, 8), baritone_one_musicmaker_three],
2612     [(196, 8), (198, 8), baritone_one_musicmaker_three],
2613     [(198, 8), (199, 8), baritone_one_musicmaker_three],
2614     [(199, 8), (200, 8), baritone_one_musicmaker_three],
2615   ],
2616 ]
2617
2618 voice_17_timespan_list = abjad.TimespanList([
2619     abjad.AnnotatedTimespan(
2620         start_offset=start_offset,
2621         stop_offset=stop_offset,
2622         annotation=MusicSpecifier(
2623             music_maker=music_maker,
2624             voice_name='Voice 17',
2625         ),
2626     ),
2627     for start_offset, stop_offset, music_maker in [
2628         [(0, 8), (2, 8), baritone_two_musicmaker_three],
2629         [(2, 8), (4, 8), baritone_two_musicmaker_three],
2630         [(4, 8), (6, 8), baritone_two_musicmaker_one],
2631         [(6, 8), (8, 8), baritone_two_musicmaker_two],
2632         [(8, 8), (10, 8), baritone_two_musicmaker_one],
2633         [(10, 8), (12, 8), baritone_two_musicmaker_three],
2634         [(12, 8), (14, 8), baritone_two_musicmaker_three],
2635         [(14, 8), (16, 8), baritone_two_musicmaker_three],
2636         [(16, 8), (18, 8), baritone_two_musicmaker_three],
2637         [(18, 8), (20, 8), baritone_two_musicmaker_one],

```

```
2638      [(20, 8), (22, 8), baritone_two_musicmaker_two],  
2639      [(22, 8), (24, 8), baritone_two_musicmaker_one],  
2640      [(24, 8), (26, 8), baritone_two_musicmaker_one],  
2641      [(26, 8), (28, 8), baritone_two_musicmaker_three],  
2642      [(28, 8), (30, 8), baritone_two_musicmaker_three],  
2643      [(30, 8), (32, 8), baritone_two_musicmaker_three],  
2644      [(32, 8), (34, 8), baritone_two_musicmaker_three],  
2645      [(34, 8), (36, 8), baritone_two_musicmaker_three],  
2646      [(36, 8), (38, 8), baritone_two_musicmaker_three],  
2647      [(38, 8), (40, 8), baritone_two_musicmaker_three],  
2648      [(40, 8), (42, 8), baritone_two_musicmaker_two],  
2649      [(42, 8), (44, 8), baritone_two_musicmaker_one],  
2650      [(44, 8), (46, 8), baritone_two_musicmaker_one],  
2651      [(46, 8), (48, 8), baritone_two_musicmaker_three],  
2652      [(48, 8), (50, 8), baritone_two_musicmaker_three],  
2653      [(50, 8), (52, 8), baritone_two_musicmaker_one],  
2654      [(52, 8), (54, 8), baritone_two_musicmaker_one],  
2655      [(54, 8), (56, 8), baritone_two_musicmaker_three],  
2656      [(56, 8), (58, 8), baritone_two_musicmaker_three],  
2657      [(58, 8), (60, 8), baritone_two_musicmaker_three],  
2658      [(60, 8), (62, 8), baritone_two_musicmaker_three],  
2659      [(62, 8), (64, 8), baritone_two_musicmaker_three],  
2660      [(64, 8), (66, 8), baritone_two_musicmaker_three],  
2661      [(66, 8), (68, 8), baritone_two_musicmaker_three],  
2662      [(68, 8), (70, 8), baritone_two_musicmaker_three],  
2663      [(70, 8), (72, 8), baritone_two_musicmaker_two],  
2664      [(72, 8), (74, 8), baritone_two_musicmaker_one],  
2665      [(74, 8), (76, 8), baritone_two_musicmaker_two],  
2666      [(76, 8), (78, 8), baritone_two_musicmaker_one],  
2667      [(78, 8), (80, 8), baritone_two_musicmaker_one],  
2668      [(80, 8), (82, 8), baritone_two_musicmaker_two],  
2669      [(82, 8), (84, 8), baritone_two_musicmaker_three],  
2670      [(84, 8), (86, 8), baritone_two_musicmaker_three],  
2671      [(86, 8), (88, 8), baritone_two_musicmaker_three],  
2672      [(88, 8), (90, 8), baritone_two_musicmaker_three],  
2673      [(90, 8), (92, 8), baritone_two_musicmaker_three],  
2674      [(92, 8), (94, 8), baritone_two_musicmaker_three],  
2675      [(94, 8), (96, 8), baritone_two_musicmaker_two],  
2676      [(96, 8), (98, 8), baritone_two_musicmaker_two],  
2677      [(98, 8), (100, 8), baritone_two_musicmaker_one],  
2678      [(100, 8), (102, 8), baritone_two_musicmaker_one],  
2679      [(102, 8), (104, 8), baritone_two_musicmaker_two],  
2680      [(104, 8), (106, 8), baritone_two_musicmaker_one],  
2681      [(106, 8), (108, 8), baritone_two_musicmaker_two],  
2682      [(108, 8), (110, 8), baritone_two_musicmaker_three],  
2683      [(110, 8), (112, 8), baritone_two_musicmaker_three],  
2684      [(112, 8), (114, 8), baritone_two_musicmaker_three],  
2685      [(114, 8), (116, 8), baritone_two_musicmaker_three],  
2686      [(116, 8), (118, 8), baritone_two_musicmaker_three],  
2687      [(118, 8), (120, 8), baritone_two_musicmaker_one],  
2688      [(120, 8), (122, 8), baritone_two_musicmaker_three],  
2689      [(122, 8), (124, 8), baritone_two_musicmaker_three],  
2690      [(124, 8), (126, 8), baritone_two_musicmaker_one],  
2691      [(126, 8), (128, 8), baritone_two_musicmaker_two],  
2692      [(128, 8), (130, 8), baritone_two_musicmaker_two],  
2693      [(130, 8), (132, 8), baritone_two_musicmaker_three],  
2694      [(132, 8), (134, 8), baritone_two_musicmaker_three],  
2695      [(134, 8), (136, 8), baritone_two_musicmaker_three],  
2696      [(136, 8), (138, 8), baritone_two_musicmaker_two],
```

```

2697     [(138, 8), (140, 8), baritone_two_musicmaker_one],
2698     [(140, 8), (142, 8), baritone_two_musicmaker_one],
2699     [(142, 8), (144, 8), baritone_two_musicmaker_three],
2700     [(144, 8), (146, 8), baritone_two_musicmaker_three],
2701     [(146, 8), (148, 8), baritone_two_musicmaker_three],
2702     [(148, 8), (150, 8), baritone_two_musicmaker_three],
2703     [(150, 8), (152, 8), baritone_two_musicmaker_three],
2704     [(152, 8), (154, 8), baritone_two_musicmaker_three],
2705     [(154, 8), (156, 8), baritone_two_musicmaker_three],
2706     [(156, 8), (158, 8), baritone_two_musicmaker_three],
2707     [(158, 8), (160, 8), baritone_two_musicmaker_three],
2708     [(160, 8), (162, 8), baritone_two_musicmaker_one],
2709     [(162, 8), (164, 8), baritone_two_musicmaker_two],
2710     [(164, 8), (166, 8), baritone_two_musicmaker_two],
2711     [(166, 8), (168, 8), baritone_two_musicmaker_two],
2712     [(168, 8), (170, 8), baritone_two_musicmaker_one],
2713     [(170, 8), (172, 8), baritone_two_musicmaker_three],
2714     [(172, 8), (174, 8), baritone_two_musicmaker_three],
2715     [(174, 8), (176, 8), baritone_two_musicmaker_three],
2716     [(176, 8), (178, 8), baritone_two_musicmaker_three],
2717     [(178, 8), (180, 8), baritone_two_musicmaker_three],
2718     [(180, 8), (182, 8), baritone_two_musicmaker_three],
2719     [(182, 8), (184, 8), baritone_two_musicmaker_one],
2720     [(184, 8), (186, 8), baritone_two_musicmaker_two],
2721     [(186, 8), (188, 8), baritone_two_musicmaker_three],
2722     [(188, 8), (190, 8), baritone_two_musicmaker_three],
2723     [(190, 8), (192, 8), baritone_two_musicmaker_three],
2724     [(192, 8), (194, 8), baritone_two_musicmaker_three],
2725     [(194, 8), (196, 8), baritone_two_musicmaker_one],
2726     [(196, 8), (198, 8), baritone_two_musicmaker_one],
2727     [(198, 8), (199, 8), baritone_two_musicmaker_three],
2728     [(199, 8), (200, 8), baritone_two_musicmaker_three],
2729   ]
2730 ])
2731
2732 voice_18_timespan_list = abjad.TimespanList([
2733     abjad.AnnotatedTimespan(
2734         start_offset=start_offset,
2735         stop_offset=stop_offset,
2736         annotation=MusicSpecifier(
2737             music_maker=music_maker,
2738             voice_name='Voice 18',
2739         ),
2740     )
2741     for start_offset, stop_offset, music_maker in [
2742         [(0, 8), (2, 8), baritone_three_musicmaker_three],
2743         [(2, 8), (4, 8), baritone_three_musicmaker_three],
2744         [(4, 8), (6, 8), baritone_three_musicmaker_three],
2745         [(6, 8), (8, 8), baritone_three_musicmaker_three],
2746         [(8, 8), (10, 8), baritone_three_musicmaker_one],
2747         [(10, 8), (12, 8), baritone_three_musicmaker_one],
2748         [(12, 8), (14, 8), baritone_three_musicmaker_two],
2749         [(14, 8), (16, 8), baritone_three_musicmaker_one],
2750         [(16, 8), (18, 8), baritone_three_musicmaker_three],
2751         [(18, 8), (20, 8), baritone_three_musicmaker_three],
2752         [(20, 8), (22, 8), baritone_three_musicmaker_three],
2753         [(22, 8), (24, 8), baritone_three_musicmaker_three],
2754         [(24, 8), (26, 8), baritone_three_musicmaker_three],
2755         [(26, 8), (28, 8), baritone_three_musicmaker_one],

```

```
2756      [(28, 8), (30, 8), baritone_three_musicmaker_two],  
2757      [(30, 8), (32, 8), baritone_three_musicmaker_one],  
2758      [(32, 8), (34, 8), baritone_three_musicmaker_two],  
2759      [(34, 8), (36, 8), baritone_three_musicmaker_three],  
2760      [(36, 8), (38, 8), baritone_three_musicmaker_three],  
2761      [(38, 8), (40, 8), baritone_three_musicmaker_three],  
2762      [(40, 8), (42, 8), baritone_three_musicmaker_one],  
2763      [(42, 8), (44, 8), baritone_three_musicmaker_three],  
2764      [(44, 8), (46, 8), baritone_three_musicmaker_three],  
2765      [(46, 8), (48, 8), baritone_three_musicmaker_one],  
2766      [(48, 8), (50, 8), baritone_three_musicmaker_three],  
2767      [(50, 8), (52, 8), baritone_three_musicmaker_three],  
2768      [(52, 8), (54, 8), baritone_three_musicmaker_three],  
2769      [(54, 8), (56, 8), baritone_three_musicmaker_three],  
2770      [(56, 8), (58, 8), baritone_three_musicmaker_two],  
2771      [(58, 8), (60, 8), baritone_three_musicmaker_one],  
2772      [(60, 8), (62, 8), baritone_three_musicmaker_one],  
2773      [(62, 8), (64, 8), baritone_three_musicmaker_three],  
2774      [(64, 8), (66, 8), baritone_three_musicmaker_three],  
2775      [(66, 8), (68, 8), baritone_three_musicmaker_three],  
2776      [(68, 8), (70, 8), baritone_three_musicmaker_three],  
2777      [(70, 8), (72, 8), baritone_three_musicmaker_two],  
2778      [(72, 8), (74, 8), baritone_three_musicmaker_one],  
2779      [(74, 8), (76, 8), baritone_three_musicmaker_two],  
2780      [(76, 8), (78, 8), baritone_three_musicmaker_one],  
2781      [(78, 8), (80, 8), baritone_three_musicmaker_one],  
2782      [(80, 8), (82, 8), baritone_three_musicmaker_three],  
2783      [(82, 8), (84, 8), baritone_three_musicmaker_three],  
2784      [(84, 8), (86, 8), baritone_three_musicmaker_three],  
2785      [(86, 8), (88, 8), baritone_three_musicmaker_two],  
2786      [(88, 8), (90, 8), baritone_three_musicmaker_one],  
2787      [(90, 8), (92, 8), baritone_three_musicmaker_one],  
2788      [(92, 8), (94, 8), baritone_three_musicmaker_three],  
2789      [(94, 8), (96, 8), baritone_three_musicmaker_three],  
2790      [(96, 8), (98, 8), baritone_three_musicmaker_three],  
2791      [(98, 8), (100, 8), baritone_three_musicmaker_three],  
2792      [(100, 8), (102, 8), baritone_three_musicmaker_three],  
2793      [(102, 8), (104, 8), baritone_three_musicmaker_three],  
2794      [(104, 8), (106, 8), baritone_three_musicmaker_two],  
2795      [(106, 8), (108, 8), baritone_three_musicmaker_one],  
2796      [(108, 8), (110, 8), baritone_three_musicmaker_one],  
2797      [(110, 8), (112, 8), baritone_three_musicmaker_two],  
2798      [(112, 8), (114, 8), baritone_three_musicmaker_two],  
2799      [(114, 8), (116, 8), baritone_three_musicmaker_two],  
2800      [(116, 8), (118, 8), baritone_three_musicmaker_three],  
2801      [(118, 8), (120, 8), baritone_three_musicmaker_three],  
2802      [(120, 8), (122, 8), baritone_three_musicmaker_three],  
2803      [(122, 8), (124, 8), baritone_three_musicmaker_three],  
2804      [(124, 8), (126, 8), baritone_three_musicmaker_three],  
2805      [(126, 8), (128, 8), baritone_three_musicmaker_one],  
2806      [(128, 8), (130, 8), baritone_three_musicmaker_two],  
2807      [(130, 8), (132, 8), baritone_three_musicmaker_one],  
2808      [(132, 8), (134, 8), baritone_three_musicmaker_three],  
2809      [(134, 8), (136, 8), baritone_three_musicmaker_three],  
2810      [(136, 8), (138, 8), baritone_three_musicmaker_three],  
2811      [(138, 8), (140, 8), baritone_three_musicmaker_three],  
2812      [(140, 8), (142, 8), baritone_three_musicmaker_three],  
2813      [(142, 8), (144, 8), baritone_three_musicmaker_three],  
2814      [(144, 8), (146, 8), baritone_three_musicmaker_three],
```

```

2815     [(146, 8), (148, 8), baritone_three_musicmaker_one],
2816     [(148, 8), (150, 8), baritone_three_musicmaker_one],
2817     [(150, 8), (152, 8), baritone_three_musicmaker_two],
2818     [(152, 8), (154, 8), baritone_three_musicmaker_two],
2819     [(154, 8), (156, 8), baritone_three_musicmaker_three],
2820     [(156, 8), (158, 8), baritone_three_musicmaker_three],
2821     [(158, 8), (160, 8), baritone_three_musicmaker_three],
2822     [(160, 8), (162, 8), baritone_three_musicmaker_three],
2823     [(162, 8), (164, 8), baritone_three_musicmaker_two],
2824     [(164, 8), (166, 8), baritone_three_musicmaker_one],
2825     [(166, 8), (168, 8), baritone_three_musicmaker_two],
2826     [(168, 8), (170, 8), baritone_three_musicmaker_one],
2827     [(170, 8), (172, 8), baritone_three_musicmaker_two],
2828     [(172, 8), (174, 8), baritone_three_musicmaker_one],
2829     [(174, 8), (176, 8), baritone_three_musicmaker_three],
2830     [(176, 8), (178, 8), baritone_three_musicmaker_three],
2831     [(178, 8), (180, 8), baritone_three_musicmaker_three],
2832     [(180, 8), (182, 8), baritone_three_musicmaker_three],
2833     [(182, 8), (184, 8), baritone_three_musicmaker_three],
2834     [(184, 8), (186, 8), baritone_three_musicmaker_three],
2835     [(186, 8), (188, 8), baritone_three_musicmaker_one],
2836     [(188, 8), (190, 8), baritone_three_musicmaker_two],
2837     [(190, 8), (192, 8), baritone_three_musicmaker_two],
2838     [(192, 8), (194, 8), baritone_three_musicmaker_one],
2839     [(194, 8), (196, 8), baritone_three_musicmaker_three],
2840     [(196, 8), (198, 8), baritone_three_musicmaker_three],
2841     [(198, 8), (199, 8), baritone_three_musicmaker_one],
2842     [(199, 8), (200, 8), baritone_three_musicmaker_three],
2843   ]
2844 ])
2845
2846 voice_19_timespan_list = abjad.TimespanList([
2847     abjad.AnnotatedTimespan(
2848         start_offset=start_offset,
2849         stop_offset=stop_offset,
2850         annotation=MusicSpecifier(
2851             music_maker=music_maker,
2852             voice_name='Voice 19',
2853         ),
2854     )
2855     for start_offset, stop_offset, music_maker in [
2856         [(0, 8), (2, 8), bass_one_musicmaker_one],
2857         [(2, 8), (4, 8), bass_one_musicmaker_two],
2858         [(4, 8), (6, 8), bass_one_musicmaker_one],
2859         [(6, 8), (8, 8), bass_one_musicmaker_three],
2860         [(8, 8), (10, 8), bass_one_musicmaker_three],
2861         [(10, 8), (12, 8), bass_one_musicmaker_three],
2862         [(12, 8), (14, 8), bass_one_musicmaker_three],
2863         [(14, 8), (16, 8), bass_one_musicmaker_one],
2864         [(16, 8), (18, 8), bass_one_musicmaker_three],
2865         [(18, 8), (20, 8), bass_one_musicmaker_three],
2866         [(20, 8), (22, 8), bass_one_musicmaker_three],
2867         [(22, 8), (24, 8), bass_one_musicmaker_one],
2868         [(24, 8), (26, 8), bass_one_musicmaker_two],
2869         [(26, 8), (28, 8), bass_one_musicmaker_one],
2870         [(28, 8), (30, 8), bass_one_musicmaker_one],
2871         [(30, 8), (32, 8), bass_one_musicmaker_three],
2872         [(32, 8), (34, 8), bass_one_musicmaker_three],
2873         [(34, 8), (36, 8), bass_one_musicmaker_three],

```

```
2874      [(36, 8), (38, 8), bass_one_musicmaker_three],  
2875      [(38, 8), (40, 8), bass_one_musicmaker_three],  
2876      [(40, 8), (42, 8), bass_one_musicmaker_three],  
2877      [(42, 8), (44, 8), bass_one_musicmaker_three],  
2878      [(44, 8), (46, 8), bass_one_musicmaker_three],  
2879      [(46, 8), (48, 8), bass_one_musicmaker_three],  
2880      [(48, 8), (50, 8), bass_one_musicmaker_three],  
2881      [(50, 8), (52, 8), bass_one_musicmaker_one],  
2882      [(52, 8), (54, 8), bass_one_musicmaker_one],  
2883      [(54, 8), (56, 8), bass_one_musicmaker_two],  
2884      [(56, 8), (58, 8), bass_one_musicmaker_one],  
2885      [(58, 8), (60, 8), bass_one_musicmaker_two],  
2886      [(60, 8), (62, 8), bass_one_musicmaker_two],  
2887      [(62, 8), (64, 8), bass_one_musicmaker_two],  
2888      [(64, 8), (68, 8), bass_one_musicmaker_three],  
2889      [(68, 8), (70, 8), bass_one_musicmaker_three],  
2890      [(70, 8), (72, 8), bass_one_musicmaker_three],  
2891      [(72, 8), (74, 8), bass_one_musicmaker_three],  
2892      [(74, 8), (76, 8), bass_one_musicmaker_three],  
2893      [(76, 8), (78, 8), bass_one_musicmaker_two],  
2894      [(78, 8), (80, 8), bass_one_musicmaker_one],  
2895      [(80, 8), (82, 8), bass_one_musicmaker_one],  
2896      [(82, 8), (84, 8), bass_one_musicmaker_three],  
2897      [(84, 8), (86, 8), bass_one_musicmaker_three],  
2898      [(86, 8), (88, 8), bass_one_musicmaker_one],  
2899      [(88, 8), (90, 8), bass_one_musicmaker_two],  
2900      [(90, 8), (92, 8), bass_one_musicmaker_three],  
2901      [(92, 8), (94, 8), bass_one_musicmaker_three],  
2902      [(94, 8), (96, 8), bass_one_musicmaker_three],  
2903      [(96, 8), (98, 8), bass_one_musicmaker_three],  
2904      [(98, 8), (100, 8), bass_one_musicmaker_two],  
2905      [(100, 8), (102, 8), bass_one_musicmaker_one],  
2906      [(102, 8), (104, 8), bass_one_musicmaker_one],  
2907      [(104, 8), (106, 8), bass_one_musicmaker_three],  
2908      [(106, 8), (108, 8), bass_one_musicmaker_three],  
2909      [(108, 8), (110, 8), bass_one_musicmaker_one],  
2910      [(110, 8), (112, 8), bass_one_musicmaker_two],  
2911      [(112, 8), (114, 8), bass_one_musicmaker_one],  
2912      [(114, 8), (116, 8), bass_one_musicmaker_two],  
2913      [(116, 8), (118, 8), bass_one_musicmaker_two],  
2914      [(118, 8), (120, 8), bass_one_musicmaker_three],  
2915      [(120, 8), (122, 8), bass_one_musicmaker_three],  
2916      [(122, 8), (124, 8), bass_one_musicmaker_three],  
2917      [(124, 8), (126, 8), bass_one_musicmaker_three],  
2918      [(126, 8), (128, 8), bass_one_musicmaker_three],  
2919      [(128, 8), (130, 8), bass_one_musicmaker_three],  
2920      [(130, 8), (132, 8), bass_one_musicmaker_three],  
2921      [(132, 8), (134, 8), bass_one_musicmaker_three],  
2922      [(134, 8), (136, 8), bass_one_musicmaker_two],  
2923      [(136, 8), (138, 8), bass_one_musicmaker_one],  
2924      [(138, 8), (140, 8), bass_one_musicmaker_two],  
2925      [(140, 8), (142, 8), bass_one_musicmaker_three],  
2926      [(142, 8), (144, 8), bass_one_musicmaker_three],  
2927      [(144, 8), (146, 8), bass_one_musicmaker_three],  
2928      [(146, 8), (148, 8), bass_one_musicmaker_three],  
2929      [(148, 8), (150, 8), bass_one_musicmaker_one],  
2930      [(150, 8), (152, 8), bass_one_musicmaker_one],  
2931      [(152, 8), (154, 8), bass_one_musicmaker_two],  
2932      [(154, 8), (156, 8), bass_one_musicmaker_two],
```

```

2933     [(156, 8), (158, 8), bass_one_musicmaker_one],
2934     [(158, 8), (160, 8), bass_one_musicmaker_three],
2935     [(160, 8), (162, 8), bass_one_musicmaker_three],
2936     [(162, 8), (164, 8), bass_one_musicmaker_three],
2937     [(164, 8), (168, 8), bass_one_musicmaker_three],
2938     [(168, 8), (170, 8), bass_one_musicmaker_one],
2939     [(170, 8), (172, 8), bass_one_musicmaker_one],
2940     [(172, 8), (174, 8), bass_one_musicmaker_two],
2941     [(174, 8), (176, 8), bass_one_musicmaker_one],
2942     [(176, 8), (178, 8), bass_one_musicmaker_three],
2943     [(178, 8), (180, 8), bass_one_musicmaker_three],
2944     [(180, 8), (182, 8), bass_one_musicmaker_three],
2945     [(182, 8), (184, 8), bass_one_musicmaker_three],
2946     [(184, 8), (186, 8), bass_one_musicmaker_three],
2947     [(186, 8), (188, 8), bass_one_musicmaker_three],
2948     [(188, 8), (190, 8), bass_one_musicmaker_three],
2949     [(190, 8), (192, 8), bass_one_musicmaker_two],
2950     [(192, 8), (194, 8), bass_one_musicmaker_two],
2951     [(194, 8), (196, 8), bass_one_musicmaker_two],
2952     [(196, 8), (198, 8), bass_one_musicmaker_one],
2953     [(198, 8), (199, 8), bass_one_musicmaker_one],
2954     [(199, 8), (200, 8), bass_one_musicmaker_three],
2955 ]
2956 ])
2957
2958 voice_20_timespan_list = abjad.TimespanList([
2959     abjad.AnnotatedTimespan(
2960         start_offset=start_offset,
2961         stop_offset=stop_offset,
2962         annotation=MusicSpecifier(
2963             music_maker=music_maker,
2964             voice_name='Voice 20',
2965         ),
2966     ),
2967     for start_offset, stop_offset, music_maker in [
2968         [(0, 8), (2, 8), bass_two_musicmaker_one],
2969         [(2, 8), (4, 8), bass_two_musicmaker_one],
2970         [(4, 8), (6, 8), bass_two_musicmaker_three],
2971         [(6, 8), (8, 8), bass_two_musicmaker_three],
2972         [(8, 8), (10, 8), bass_two_musicmaker_three],
2973         [(10, 8), (12, 8), bass_two_musicmaker_three],
2974         [(12, 8), (14, 8), bass_two_musicmaker_one],
2975         [(14, 8), (16, 8), bass_two_musicmaker_one],
2976         [(16, 8), (18, 8), bass_two_musicmaker_two],
2977         [(18, 8), (20, 8), bass_two_musicmaker_three],
2978         [(20, 8), (22, 8), bass_two_musicmaker_three],
2979         [(22, 8), (24, 8), bass_two_musicmaker_three],
2980         [(24, 8), (26, 8), bass_two_musicmaker_one],
2981         [(26, 8), (28, 8), bass_two_musicmaker_one],
2982         [(28, 8), (30, 8), bass_two_musicmaker_two],
2983         [(30, 8), (32, 8), bass_two_musicmaker_two],
2984         [(32, 8), (34, 8), bass_two_musicmaker_three],
2985         [(34, 8), (36, 8), bass_two_musicmaker_three],
2986         [(36, 8), (38, 8), bass_two_musicmaker_three],
2987         [(38, 8), (40, 8), bass_two_musicmaker_three],
2988         [(40, 8), (42, 8), bass_two_musicmaker_three],
2989         [(42, 8), (44, 8), bass_two_musicmaker_two],
2990         [(44, 8), (46, 8), bass_two_musicmaker_two],
2991         [(46, 8), (48, 8), bass_two_musicmaker_one],

```

```
2992      [(48, 8), (50, 8), bass_two_musicmaker_two],  
2993      [(50, 8), (52, 8), bass_two_musicmaker_one],  
2994      [(52, 8), (54, 8), bass_two_musicmaker_one],  
2995      [(54, 8), (56, 8), bass_two_musicmaker_two],  
2996      [(56, 8), (58, 8), bass_two_musicmaker_one],  
2997      [(58, 8), (60, 8), bass_two_musicmaker_two],  
2998      [(60, 8), (62, 8), bass_two_musicmaker_three],  
2999      [(62, 8), (64, 8), bass_two_musicmaker_three],  
3000      [(64, 8), (68, 8), bass_two_musicmaker_three],  
3001      [(68, 8), (70, 8), bass_two_musicmaker_three],  
3002      [(70, 8), (72, 8), bass_two_musicmaker_three],  
3003      [(72, 8), (74, 8), bass_two_musicmaker_three],  
3004      [(74, 8), (76, 8), bass_two_musicmaker_three],  
3005      [(76, 8), (78, 8), bass_two_musicmaker_three],  
3006      [(78, 8), (80, 8), bass_two_musicmaker_two],  
3007      [(80, 8), (82, 8), bass_two_musicmaker_one],  
3008      [(82, 8), (84, 8), bass_two_musicmaker_two],  
3009      [(84, 8), (86, 8), bass_two_musicmaker_one],  
3010      [(86, 8), (88, 8), bass_two_musicmaker_one],  
3011      [(88, 8), (90, 8), bass_two_musicmaker_two],  
3012      [(90, 8), (92, 8), bass_two_musicmaker_three],  
3013      [(92, 8), (94, 8), bass_two_musicmaker_three],  
3014      [(94, 8), (96, 8), bass_two_musicmaker_three],  
3015      [(96, 8), (98, 8), bass_two_musicmaker_three],  
3016      [(98, 8), (100, 8), bass_two_musicmaker_three],  
3017      [(100, 8), (102, 8), bass_two_musicmaker_one],  
3018      [(102, 8), (104, 8), bass_two_musicmaker_two],  
3019      [(104, 8), (106, 8), bass_two_musicmaker_two],  
3020      [(106, 8), (108, 8), bass_two_musicmaker_one],  
3021      [(108, 8), (110, 8), bass_two_musicmaker_one],  
3022      [(110, 8), (112, 8), bass_two_musicmaker_two],  
3023      [(112, 8), (114, 8), bass_two_musicmaker_one],  
3024      [(114, 8), (116, 8), bass_two_musicmaker_one],  
3025      [(116, 8), (118, 8), bass_two_musicmaker_three],  
3026      [(118, 8), (120, 8), bass_two_musicmaker_three],  
3027      [(120, 8), (122, 8), bass_two_musicmaker_three],  
3028      [(122, 8), (124, 8), bass_two_musicmaker_three],  
3029      [(124, 8), (126, 8), bass_two_musicmaker_three],  
3030      [(126, 8), (128, 8), bass_two_musicmaker_three],  
3031      [(128, 8), (130, 8), bass_two_musicmaker_three],  
3032      [(130, 8), (132, 8), bass_two_musicmaker_three],  
3033      [(132, 8), (134, 8), bass_two_musicmaker_three],  
3034      [(134, 8), (136, 8), bass_two_musicmaker_three],  
3035      [(136, 8), (138, 8), bass_two_musicmaker_two],  
3036      [(138, 8), (140, 8), bass_two_musicmaker_one],  
3037      [(140, 8), (142, 8), bass_two_musicmaker_two],  
3038      [(142, 8), (144, 8), bass_two_musicmaker_three],  
3039      [(144, 8), (146, 8), bass_two_musicmaker_three],  
3040      [(146, 8), (148, 8), bass_two_musicmaker_one],  
3041      [(148, 8), (150, 8), bass_two_musicmaker_three],  
3042      [(150, 8), (152, 8), bass_two_musicmaker_three],  
3043      [(152, 8), (154, 8), bass_two_musicmaker_three],  
3044      [(154, 8), (156, 8), bass_two_musicmaker_two],  
3045      [(156, 8), (158, 8), bass_two_musicmaker_one],  
3046      [(158, 8), (160, 8), bass_two_musicmaker_one],  
3047      [(160, 8), (162, 8), bass_two_musicmaker_three],  
3048      [(162, 8), (164, 8), bass_two_musicmaker_three],  
3049      [(164, 8), (168, 8), bass_two_musicmaker_three],  
3050      [(168, 8), (170, 8), bass_two_musicmaker_three],
```

```

3051     [(170, 8), (172, 8), bass_two_musicmaker_one],
3052     [(172, 8), (174, 8), bass_two_musicmaker_one],
3053     [(174, 8), (176, 8), bass_two_musicmaker_two],
3054     [(176, 8), (178, 8), bass_two_musicmaker_two],
3055     [(178, 8), (180, 8), bass_two_musicmaker_three],
3056     [(180, 8), (182, 8), bass_two_musicmaker_three],
3057     [(182, 8), (184, 8), bass_two_musicmaker_three],
3058     [(184, 8), (186, 8), bass_two_musicmaker_three],
3059     [(186, 8), (188, 8), bass_two_musicmaker_three],
3060     [(188, 8), (190, 8), bass_two_musicmaker_three],
3061     [(190, 8), (192, 8), bass_two_musicmaker_three],
3062     [(192, 8), (194, 8), bass_two_musicmaker_three],
3063     [(194, 8), (196, 8), bass_two_musicmaker_two],
3064     [(196, 8), (198, 8), bass_two_musicmaker_one],
3065     [(198, 8), (199, 8), bass_two_musicmaker_two],
3066     [(199, 8), (200, 8), bass_two_musicmaker_two],
3067   ]
3068 ])
3069
3070 voice_21_timespan_list = abjad.TimespanList([
3071     abjad.AnnotatedTimespan(
3072         start_offset=start_offset,
3073         stop_offset=stop_offset,
3074         annotation=MusicSpecifier(
3075             music_maker=music_maker,
3076             voice_name='Voice 21',
3077         ),
3078     )
3079     for start_offset, stop_offset, music_maker in [
3080         [(0, 8), (2, 8), contrabass_musicmaker_one],
3081         [(2, 8), (4, 8), contrabass_musicmaker_one],
3082         [(4, 8), (6, 8), contrabass_musicmaker_one],
3083         [(6, 8), (8, 8), contrabass_musicmaker_three],
3084         [(8, 8), (10, 8), contrabass_musicmaker_three],
3085         [(10, 8), (12, 8), contrabass_musicmaker_three],
3086         [(12, 8), (14, 8), contrabass_musicmaker_one],
3087         [(14, 8), (16, 8), contrabass_musicmaker_two],
3088         [(16, 8), (18, 8), contrabass_musicmaker_one],
3089         [(18, 8), (20, 8), contrabass_musicmaker_three],
3090         [(20, 8), (22, 8), contrabass_musicmaker_three],
3091         [(22, 8), (24, 8), contrabass_musicmaker_one],
3092         [(24, 8), (26, 8), contrabass_musicmaker_two],
3093         [(26, 8), (28, 8), contrabass_musicmaker_one],
3094         [(28, 8), (30, 8), contrabass_musicmaker_one],
3095         [(30, 8), (32, 8), contrabass_musicmaker_two],
3096         [(32, 8), (34, 8), contrabass_musicmaker_three],
3097         [(34, 8), (36, 8), contrabass_musicmaker_three],
3098         [(36, 8), (38, 8), contrabass_musicmaker_three],
3099         [(38, 8), (40, 8), contrabass_musicmaker_three],
3100         [(40, 8), (42, 8), contrabass_musicmaker_three],
3101         [(42, 8), (44, 8), contrabass_musicmaker_three],
3102         [(44, 8), (46, 8), contrabass_musicmaker_three],
3103         [(46, 8), (48, 8), contrabass_musicmaker_two],
3104         [(48, 8), (50, 8), contrabass_musicmaker_one],
3105         [(50, 8), (52, 8), contrabass_musicmaker_one],
3106         [(52, 8), (54, 8), contrabass_musicmaker_two],
3107         [(54, 8), (56, 8), contrabass_musicmaker_one],
3108         [(56, 8), (58, 8), contrabass_musicmaker_one],
3109         [(58, 8), (60, 8), contrabass_musicmaker_three],

```

```
3110      [(60, 8), (62, 8), contrabass_musicmaker_three],  
3111      [(62, 8), (64, 8), contrabass_musicmaker_three],  
3112      [(64, 8), (68, 8), contrabass_musicmaker_three],  
3113      [(68, 8), (70, 8), contrabass_musicmaker_three],  
3114      [(70, 8), (72, 8), contrabass_musicmaker_three],  
3115      [(72, 8), (74, 8), contrabass_musicmaker_two],  
3116      [(74, 8), (76, 8), contrabass_musicmaker_one],  
3117      [(76, 8), (78, 8), contrabass_musicmaker_one],  
3118      [(78, 8), (80, 8), contrabass_musicmaker_two],  
3119      [(80, 8), (82, 8), contrabass_musicmaker_one],  
3120      [(82, 8), (84, 8), contrabass_musicmaker_one],  
3121      [(84, 8), (86, 8), contrabass_musicmaker_one],  
3122      [(86, 8), (88, 8), contrabass_musicmaker_one],  
3123      [(88, 8), (90, 8), contrabass_musicmaker_three],  
3124      [(90, 8), (92, 8), contrabass_musicmaker_three],  
3125      [(92, 8), (94, 8), contrabass_musicmaker_three],  
3126      [(94, 8), (96, 8), contrabass_musicmaker_three],  
3127      [(96, 8), (98, 8), contrabass_musicmaker_three],  
3128      [(98, 8), (100, 8), contrabass_musicmaker_three],  
3129      [(100, 8), (102, 8), contrabass_musicmaker_three],  
3130      [(102, 8), (104, 8), contrabass_musicmaker_three],  
3131      [(104, 8), (106, 8), contrabass_musicmaker_three],  
3132      [(106, 8), (108, 8), contrabass_musicmaker_one],  
3133      [(108, 8), (110, 8), contrabass_musicmaker_two],  
3134      [(110, 8), (112, 8), contrabass_musicmaker_three],  
3135      [(112, 8), (114, 8), contrabass_musicmaker_one],  
3136      [(114, 8), (116, 8), contrabass_musicmaker_one],  
3137      [(116, 8), (118, 8), contrabass_musicmaker_three],  
3138      [(118, 8), (120, 8), contrabass_musicmaker_three],  
3139      [(120, 8), (122, 8), contrabass_musicmaker_three],  
3140      [(122, 8), (124, 8), contrabass_musicmaker_three],  
3141      [(124, 8), (126, 8), contrabass_musicmaker_three],  
3142      [(126, 8), (128, 8), contrabass_musicmaker_three],  
3143      [(128, 8), (130, 8), contrabass_musicmaker_three],  
3144      [(130, 8), (132, 8), contrabass_musicmaker_two],  
3145      [(132, 8), (134, 8), contrabass_musicmaker_two],  
3146      [(134, 8), (136, 8), contrabass_musicmaker_two],  
3147      [(136, 8), (138, 8), contrabass_musicmaker_two],  
3148      [(138, 8), (140, 8), contrabass_musicmaker_three],  
3149      [(140, 8), (142, 8), contrabass_musicmaker_three],  
3150      [(142, 8), (144, 8), contrabass_musicmaker_three],  
3151      [(144, 8), (146, 8), contrabass_musicmaker_three],  
3152      [(146, 8), (148, 8), contrabass_musicmaker_three],  
3153      [(148, 8), (150, 8), contrabass_musicmaker_three],  
3154      [(150, 8), (152, 8), contrabass_musicmaker_three],  
3155      [(152, 8), (154, 8), contrabass_musicmaker_three],  
3156      [(154, 8), (156, 8), contrabass_musicmaker_two],  
3157      [(156, 8), (158, 8), contrabass_musicmaker_one],  
3158      [(158, 8), (160, 8), contrabass_musicmaker_one],  
3159      [(160, 8), (162, 8), contrabass_musicmaker_one],  
3160      [(162, 8), (164, 8), contrabass_musicmaker_one],  
3161      [(164, 8), (168, 8), contrabass_musicmaker_two],  
3162      [(168, 8), (170, 8), contrabass_musicmaker_two],  
3163      [(170, 8), (172, 8), contrabass_musicmaker_two],  
3164      [(172, 8), (174, 8), contrabass_musicmaker_one],  
3165      [(174, 8), (176, 8), contrabass_musicmaker_one],  
3166      [(176, 8), (178, 8), contrabass_musicmaker_three],  
3167      [(178, 8), (180, 8), contrabass_musicmaker_three],  
3168      [(180, 8), (182, 8), contrabass_musicmaker_three],
```

```

3169     [(182, 8), (184, 8), contrabass_musicmaker_three],
3170     [(184, 8), (186, 8), contrabass_musicmaker_three],
3171     [(186, 8), (188, 8), contrabass_musicmaker_three],
3172     [(188, 8), (190, 8), contrabass_musicmaker_three],
3173     [(190, 8), (192, 8), contrabass_musicmaker_three],
3174     [(192, 8), (194, 8), contrabass_musicmaker_two],
3175     [(194, 8), (196, 8), contrabass_musicmaker_one],
3176     [(196, 8), (198, 8), contrabass_musicmaker_three],
3177     [(198, 8), (199, 8), contrabass_musicmaker_three],
3178     [(199, 8), (200, 8), contrabass_musicmaker_three],
3179 ]
3180 ])
3181
3182 all_timespan_lists = {
3183     'Voice 1': voice_1_timespan_list,
3184     'Voice 2': voice_2_timespan_list,
3185     'Voice 3': voice_3_timespan_list,
3186     'Voice 4': voice_4_timespan_list,
3187     'Voice 5': voice_5_timespan_list,
3188     'Voice 6': voice_6_timespan_list,
3189     'Voice 7': voice_7_timespan_list,
3190     'Voice 8': voice_8_timespan_list,
3191     'Voice 9': voice_9_timespan_list,
3192     'Voice 10': voice_10_timespan_list,
3193     'Voice 11': voice_11_timespan_list,
3194     'Voice 12': voice_12_timespan_list,
3195     'Voice 13': voice_13_timespan_list,
3196     'Voice 14': voice_14_timespan_list,
3197     'Voice 15': voice_15_timespan_list,
3198     'Voice 16': voice_16_timespan_list,
3199     'Voice 17': voice_17_timespan_list,
3200     'Voice 18': voice_18_timespan_list,
3201     'Voice 19': voice_19_timespan_list,
3202     'Voice 20': voice_20_timespan_list,
3203     'Voice 21': voice_21_timespan_list,
3204 }
3205
3206 global_timespan = abjad.Timespan(
3207     start_offset=0,
3208     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values()))
3209 )
3210
3211 for voice_name, timespan_list in all_timespan_lists.items():
3212     silences = abjad.TimespanList([global_timespan])
3213     silences.extend(timespan_list)
3214     silences.sort()
3215     silences.compute_logical_xor()
3216     for silence_timespan in silences:
3217         timespan_list.append(
3218             abjad.AnnotatedTimespan(
3219                 start_offset=silence_timespan.start_offset,
3220                 stop_offset=silence_timespan.stop_offset,
3221                 annotation=MusicSpecifier(
3222                     music_maker=None,
3223                     voice_name=voice_name,
3224                     ),
3225                 )
3226             )
3227     timespan_list.sort()

```

```

3228
3229 for voice_name, timespan_list in all_timespan_lists.items():
3230     shards = timespan_list.split_at_offsets(bounds)
3231     split_timespan_list = abjad.TimespanList()
3232     for shard in shards:
3233         split_timespan_list.extend(shard)
3234     split_timespan_list.sort()
3235     all_timespan_lists[voice_name] = timespan_list
3236
3237 score = abjad.Score([
3238     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
3239     abjad.StaffGroup(
3240         [
3241             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
3242                         lilypond_type='Staff',),
3243             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
3244                         lilypond_type='Staff',),
3245             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
3246                         lilypond_type='Staff',),
3247             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
3248                         lilypond_type='Staff',),
3249             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
3250                         lilypond_type='Staff',),
3251             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
3252                         lilypond_type='Staff',),
3253             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
3254                         lilypond_type='Staff',),
3255             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
3256                         lilypond_type='Staff',),
3257             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
3258                         lilypond_type='Staff',),
3259             abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
3260                         lilypond_type='Staff',),
3261             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
3262                         lilypond_type='Staff',),
3263             abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
3264                         lilypond_type='Staff',),
3265             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
3266                         lilypond_type='Staff',),
3267             abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
3268                         lilypond_type='Staff',),
3269             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
3270                         lilypond_type='Staff',),
3271             abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
3272                         lilypond_type='Staff',),
3273             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
3274                         lilypond_type='Staff',),
3275             abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
3276                         lilypond_type='Staff',),
3277             abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
3278                         lilypond_type='Staff',),
3279             abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
3280                         lilypond_type='Staff',),
3281             abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
3282                         lilypond_type='Staff',),
3283         ],
3284         name='Staff Group',
3285     )
3286 ])

```

```

3266
3267 for time_signature in time_signatures:
3268     skip = abjad.Skip(1, multiplier=(time_signature))
3269     abjad.attach(time_signature, skip)
3270     score['Global Context'].append(skip)
3271
3272 print('Making containers ...')
3273
3274 def make_container(music_maker, durations):
3275     selections = music_maker(durations)
3276     container = abjad.Container([])
3277     container.extend(selections)
3278     return container
3279
3280 def key_function(timespan):
3281     return timespan.annotation.music_maker or silence_maker
3282
3283 for voice_name, timespan_list in all_timespan_lists.items():
3284     for music_maker, grouper in itertools.groupby(
3285         timespan_list,
3286         key=key_function,
3287     ):
3288         durations = [timespan.duration for timespan in grouper]
3289         container = make_container(music_maker, durations)
3290         voice = score[voice_name]
3291         voice.append(container)
3292
3293 print('Splitting and rewriting ...')
3294 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
3295     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
3296         time_signature = time_signatures[i]
3297         abjad.mutate(shard).rewrite_meter(time_signature)
3298
3299 trill = TrillHandler()
3300 trill(score)
3301
3302 print('Beaming runs ...')
3303 for voice in abjad.select(score).components(abjad.Voice):
3304     for run in abjad.select(voice).runs():
3305         specifier = abjadext.rmakers.BeamSpecifier(
3306             beam_each_division=False,
3307         )
3308         specifier(run)
3309         abjad.beam(voice[:, beam_lone_notes=False, beam_rests=False])
3310
3311 print('Stopping Hairpins ...')
3312 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
3313     for rest in abjad.iterate(staff).components(abjad.Rest):
3314         previous_leaf = abjad.inspect(rest).leaf(-1)
3315         if isinstance(previous_leaf, abjad.Note):
3316             abjad.attach(abjad.StopHairpin(), rest)
3317         elif isinstance(previous_leaf, abjad.Chord):
3318             abjad.attach(abjad.StopHairpin(), rest)
3319         elif isinstance(previous_leaf, abjad.Rest):
3320             pass
3321
3322 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
3323     first_leaf = abjad.select(staff).leaves()[0]
3324     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)

```

```

3325 abjad.attach(stop, first_leaf)
3326
3327 print('Adding attachments ...')
3328 bar_line = abjad.LilyPondLiteral(r'\bar "||"', format_slot='after',)
3329 metro = abjad.MetronomeMark((1, 4), 90)
3330 markup = abjad.Markup(r'\bold { E }')
3331 mark = abjad.RehearsalMark(markup=markup)
3332
3333 instruments = cyc([
3334     abjad.SopraninoSaxophone(),
3335     abjad.SopranoSaxophone(),
3336     abjad.SopranoSaxophone(),
3337     abjad.SopranoSaxophone(),
3338     abjad.AltoSaxophone(),
3339     abjad.AltoSaxophone(),
3340     abjad.AltoSaxophone(),
3341     abjad.AltoSaxophone(),
3342     abjad.AltoSaxophone(),
3343     abjad.AltoSaxophone(),
3344     abjad.TenorSaxophone(),
3345     abjad.TenorSaxophone(),
3346     abjad.TenorSaxophone(),
3347     abjad.TenorSaxophone(),
3348     abjad.TenorSaxophone(),
3349     abjad.BaritoneSaxophone(),
3350     abjad.BaritoneSaxophone(),
3351     abjad.BaritoneSaxophone(),
3352     abjad.BassSaxophone(),
3353     abjad.BassSaxophone(),
3354     abjad.ContrabassSaxophone(),
3355 ])
3356
3357 abbreviations = cyc([
3358     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
3359     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
3360     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
3361     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
3362     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
3363     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
3364     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
3365     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
3366     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
3367     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
3368     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
3369     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
3370     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
3371     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
3372     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
3373     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
3374     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
3375     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
3376     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
3377     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
3378     abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
3379 ])
3380
3381 names = cyc([
3382     abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
3383     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),

```

```

3384 abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
3385 abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
3386 abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
3387 abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
3388 abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
3389 abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
3390 abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
3391 abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
3392 abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
3393 abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
3394 abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
3395 abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
3396 abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
3397 abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
3398 abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
3399 abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
3400 abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
3401 abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
3402 abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
3403 ])
3404
3405 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
3406     leaf1 = abjad.select(staff).leaves()[0]
3407     abjad.attach(next(instruments), leaf1)
3408     abjad.attach(next(abbreviations), leaf1)
3409     abjad.attach(next(names), leaf1)
3410
3411 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
3412     leaf1 = abjad.select(staff).leaves()[0]
3413     last_leaf = abjad.select(staff).leaves()[-1]
3414     abjad.attach(metro, leaf1)
3415     abjad.attach(bar_line, last_leaf)
3416
3417 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
3418     leaf1 = abjad.select(staff).leaves()[0]
3419     abjad.attach(mark, leaf1)
3420
3421 score_file = abjad.LilyPondFile.new(
3422     score,
3423     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3424     _stylesheets/abjad.ily'],
3425 )
3426
3427 abjad.SegmentMaker.comment_measure_numbers(score)
3428 ######
3429 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_E'
3430 pdf_path = f'{directory}/Section_E.pdf'
3431 path = pathlib.Path('Section_E.pdf')
3432 if path.exists():
3433     print(f'Removing {pdf_path} ...')
3434     path.unlink()
3435 time_1 = time.time()
3436 print(f'Persisting {pdf_path} ...')
3437 result = abjad.persist(score_file).as_pdf(pdf_path)
3438 print(result[0])
3439 print(result[1])
3440 print(result[2])
3441 success = result[3]
```

```

3442 if success is False:
3443     print('LilyPond failed!')
3444 time_2 = time.time()
3445 total_time = time_2 - time_1
3446 print(f'Total time: {total_time} seconds')
3447 if path.exists():
3448     print(f'Opening {pdf_path} ...')
3449     os.system(f'open {pdf_path}')
3450 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_E/Section_E.
3451 ly').readlines()
3451 open('/Users/evansdsg2/Scores/guerrero/Build/Section_E.ly', 'w').writelines(
3452     score_lines[15:-1])
3453
3454 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
3455     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3456     signature_copy = abjad.mutate(signatures).copy()
3457     staff_copy = abjad.mutate(staff).copy()
3458     part = abjad.Score()
3459     part.insert(0, staff)
3460     part.insert(0, signature_copy)
3461     part_file = abjad.LilyPondFile.new(
3462         part,
3463         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3464         _stylesheets/abjad.ily'],
3465         )
3466     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano'
3467     pdf_path = f'{directory}/Section_E.pdf'
3468     path = pathlib.Path('Section_E.pdf')
3469     if path.exists():
3470         print(f'Removing {pdf_path} ...')
3471         path.unlink()
3472     time_1 = time.time()
3473     print(f'Persisting {pdf_path} ...')
3474     result = abjad.persist(part_file).as_pdf(pdf_path)
3475     print(result[0])
3476     print(result[1])
3477     print(result[2])
3478     success = result[3]
3479     if success is False:
3480         print('LilyPond failed!')
3481     time_2 = time.time()
3482     total_time = time_2 - time_1
3483     print(f'Total time: {total_time} seconds')
3484     if path.exists():
3485         print(f'Opening {pdf_path} ...')
3486         os.system(f'open {pdf_path}')
3487     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano/
3488     Section_E.ly').readlines()
3489     open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano/Section_E.ly',
3490         'w').writelines(part_lines[15:-1])
3491
3492 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
3493     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3494     signature_copy = abjad.mutate(signatures).copy()
3495     staff_copy = abjad.mutate(staff).copy()
3496     part = abjad.Score()
3497     part.insert(0, staff)
3498     part.insert(0, signature_copy)
3499     part_file = abjad.LilyPondFile.new(

```

```

3496     part,
3497     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3498 _stylesheets/abjad.ily'],
3499     )
3500 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1'
3501 pdf_path = f'{directory}/Section_E.pdf'
3502 path = pathlib.Path('Section_E.pdf')
3503 if path.exists():
3504     print(f'Removing {pdf_path} ...')
3505     path.unlink()
3506 time_1 = time.time()
3507 print(f'Persisting {pdf_path} ...')
3508 result = abjad.persist(part_file).as_pdf(pdf_path)
3509 print(result[0])
3510 print(result[1])
3511 print(result[2])
3512 success = result[3]
3513 if success is False:
3514     print('LilyPond failed!')
3515 time_2 = time.time()
3516 total_time = time_2 - time_1
3517 print(f'Total time: {total_time} seconds')
3518 if path.exists():
3519     print(f'Opening {pdf_path} ...')
3520     os.system(f'open {pdf_path}')
3521 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/
3522 Section_E.ly').readlines()
3523 open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/Section_E.ly',
3524 'w').writelines(part_lines[15:-1])
3525
3526 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
3527     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3528     signature_copy = abjad.mutate(signatures).copy()
3529     staff_copy = abjad.mutate(staff).copy()
3530     part = abjad.Score()
3531     part.insert(0, staff)
3532     part.insert(0, signature_copy)
3533     part_file = abjad.LilyPondFile.new(
3534         part,
3535         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3536 _stylesheets/abjad.ily'],
3537         )
3538 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
3539 pdf_path = f'{directory}/Section_E.pdf'
3540 path = pathlib.Path('Section_E.pdf')
3541 if path.exists():
3542     print(f'Removing {pdf_path} ...')
3543     path.unlink()
3544 time_1 = time.time()
3545 print(f'Persisting {pdf_path} ...')
3546 result = abjad.persist(part_file).as_pdf(pdf_path)
3547 print(result[0])
3548 print(result[1])
3549 print(result[2])
3550 success = result[3]
3551 if success is False:
3552     print('LilyPond failed!')
3553 time_2 = time.time()
3554 total_time = time_2 - time_1

```

```

3551     print(f'Total time: {total_time} seconds')
3552     if path.exists():
3553         print(f'Opening {pdf_path} ...')
3554         os.system(f'open {pdf_path}')
3555     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/
3556     Section_E.ly').readlines()
3557     open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/Section_E.ly',
3558     'w').writelines(part_lines[15:-1])
3559
3560 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
3561     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3562     signature_copy = abjad.mutate(signatures).copy()
3563     staff_copy = abjad.mutate(staff).copy()
3564     part = abjad.Score()
3565     part.insert(0, staff)
3566     part.insert(0, signature_copy)
3567     part_file = abjad.LilyPondFile.new(
3568         part,
3569         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3570         _stylesheets/abjad.ily'],
3571         )
3572     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3'
3573     pdf_path = f'{directory}/Section_E.pdf'
3574     path = pathlib.Path('Section_E.pdf')
3575     if path.exists():
3576         print(f'Removing {pdf_path} ...')
3577         path.unlink()
3578     time_1 = time.time()
3579     print(f'Persisting {pdf_path} ...')
3580     result = abjad.persist(part_file).as_pdf(pdf_path)
3581     print(result[0])
3582     print(result[1])
3583     print(result[2])
3584     success = result[3]
3585     if success is False:
3586         print('LilyPond failed!')
3587     time_2 = time.time()
3588     total_time = time_2 - time_1
3589     print(f'Total time: {total_time} seconds')
3590     if path.exists():
3591         print(f'Opening {pdf_path} ...')
3592         os.system(f'open {pdf_path}')
3593     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/
3594     Section_E.ly').readlines()
3595     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/Section_E.ly',
3596     'w').writelines(part_lines[15:-1])
3597
3598 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
3599     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3600     signature_copy = abjad.mutate(signatures).copy()
3601     staff_copy = abjad.mutate(staff).copy()
3602     part = abjad.Score()
3603     part.insert(0, staff)
3604     part.insert(0, signature_copy)
3605     part_file = abjad.LilyPondFile.new(
3606         part,
3607         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3608         _stylesheets/abjad.ily'],
3609         )

```

```

3604     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1'
3605     pdf_path = f'{directory}/Section_E.pdf'
3606     path = pathlib.Path('Section_E.pdf')
3607     if path.exists():
3608         print(f'Removing {pdf_path} ...')
3609         path.unlink()
3610     time_1 = time.time()
3611     print(f'Persisting {pdf_path} ...')
3612     result = abjad.persist(part_file).as_pdf(pdf_path)
3613     print(result[0])
3614     print(result[1])
3615     print(result[2])
3616     success = result[3]
3617     if success is False:
3618         print('LilyPond failed!')
3619     time_2 = time.time()
3620     total_time = time_2 - time_1
3621     print(f'Total time: {total_time} seconds')
3622     if path.exists():
3623         print(f'Opening {pdf_path} ...')
3624         os.system(f'open {pdf_path}')
3625     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/
3626     Section_E.ly').readlines()
3627     open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/Section_E.ly', 'w'
3628     ).writelines(part_lines[15:-1])
3629
3630 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
3631     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3632     signature_copy = abjad.mutate(signatures).copy()
3633     staff_copy = abjad.mutate(staff).copy()
3634     part = abjad.Score()
3635     part.insert(0, staff)
3636     part.insert(0, signature_copy)
3637     part_file = abjad.LilyPondFile.new(
3638         part,
3639         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3640         _stylesheets/abjad.ily'],
3641         )
3642     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
3643     pdf_path = f'{directory}/Section_E.pdf'
3644     path = pathlib.Path('Section_E.pdf')
3645     if path.exists():
3646         print(f'Removing {pdf_path} ...')
3647         path.unlink()
3648     time_1 = time.time()
3649     print(f'Persisting {pdf_path} ...')
3650     result = abjad.persist(part_file).as_pdf(pdf_path)
3651     print(result[0])
3652     print(result[1])
3653     print(result[2])
3654     success = result[3]
3655     if success is False:
3656         print('LilyPond failed!')
3657     time_2 = time.time()
3658     total_time = time_2 - time_1
3659     print(f'Total time: {total_time} seconds')

```

```

3660 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
3661 Section_E.ly').readlines()
3662 open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_E.ly', 'w'
3663 ).writelines(part_lines[15:-1])
3664
3665 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
3666 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3667 signature_copy = abjad.mutate(signatures).copy()
3668 staff_copy = abjad.mutate(staff).copy()
3669 part = abjad.Score()
3670 part.insert(0, staff)
3671 part.insert(0, signature_copy)
3672 part_file = abjad.LilyPondFile.new(
3673     part,
3674     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3675 _stylesheets/abjad.ily'],
3676 )
3677 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
3678 pdf_path = f'{directory}/Section_E.pdf'
3679 path = pathlib.Path('Section_E.pdf')
3680 if path.exists():
3681     print(f'Removing {pdf_path} ...')
3682     path.unlink()
3683 time_1 = time.time()
3684 print(f'Persisting {pdf_path} ...')
3685 result = abjad.persist(part_file).as_pdf(pdf_path)
3686 print(result[0])
3687 print(result[1])
3688 print(result[2])
3689 success = result[3]
3690 if success is False:
3691     print('LilyPond failed!')
3692 time_2 = time.time()
3693 total_time = time_2 - time_1
3694 print(f'Total time: {total_time} seconds')
3695 if path.exists():
3696     print(f'Opening {pdf_path} ...')
3697     os.system(f'open {pdf_path}')
3698 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/
3699 Section_E.ly').readlines()
3700 open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/Section_E.ly', 'w'
3701 ).writelines(part_lines[15:-1])
3702
3703 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
3704 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3705 signature_copy = abjad.mutate(signatures).copy()
3706 staff_copy = abjad.mutate(staff).copy()
3707 part = abjad.Score()
3708 part.insert(0, staff)
3709 part.insert(0, signature_copy)
3710 part_file = abjad.LilyPondFile.new(
3711     part,
3712     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3713 _stylesheets/abjad.ily'],
3714 )
3715 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
3716 pdf_path = f'{directory}/Section_E.pdf'
3717 path = pathlib.Path('Section_E.pdf')
3718 if path.exists():

```

```

3713     print(f'Removing {pdf_path} ...')
3714     path.unlink()
3715     time_1 = time.time()
3716     print(f'Persisting {pdf_path} ...')
3717     result = abjad.persist(part_file).as_pdf(pdf_path)
3718     print(result[0])
3719     print(result[1])
3720     print(result[2])
3721     success = result[3]
3722     if success is False:
3723         print('LilyPond failed!')
3724     time_2 = time.time()
3725     total_time = time_2 - time_1
3726     print(f'Total time: {total_time} seconds')
3727     if path.exists():
3728         print(f'Opening {pdf_path} ...')
3729         os.system(f'open {pdf_path}')
3730     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/
3731     Section_E.ly').readlines()
3732     open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/Section_E.ly', 'w'
3733     ).writelines(part_lines[15:-1])
3734
3735 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
3736     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3737     signature_copy = abjad.mutate(signatures).copy()
3738     staff_copy = abjad.mutate(staff).copy()
3739     part = abjad.Score()
3740     part.insert(0, staff)
3741     part.insert(0, signature_copy)
3742     part_file = abjad.LilyPondFile.new(
3743         part,
3744         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3745         _stylesheets/abjad.ily'],
3746         )
3747     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
3748     pdf_path = f'{directory}/Section_E.pdf'
3749     path = pathlib.Path('Section_E.pdf')
3750     if path.exists():
3751         print(f'Removing {pdf_path} ...')
3752         path.unlink()
3753     time_1 = time.time()
3754     print(f'Persisting {pdf_path} ...')
3755     result = abjad.persist(part_file).as_pdf(pdf_path)
3756     print(result[0])
3757     print(result[1])
3758     print(result[2])
3759     success = result[3]
3760     if success is False:
3761         print('LilyPond failed!')
3762     time_2 = time.time()
3763     total_time = time_2 - time_1
3764     print(f'Total time: {total_time} seconds')
3765     if path.exists():
3766         print(f'Opening {pdf_path} ...')
3767         os.system(f'open {pdf_path}')
3768     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/
3769     Section_E.ly').readlines()
3770     open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/Section_E.ly', 'w'
3771     ).writelines(part_lines[15:-1])

```

```

3767
3768 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
3769     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3770     signature_copy = abjad.mutate(signatures).copy()
3771     staff_copy = abjad.mutate(staff).copy()
3772     part = abjad.Score()
3773     part.insert(0, staff)
3774     part.insert(0, signature_copy)
3775     part_file = abjad.LilyPondFile.new(
3776         part,
3777         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3778 _stylesheets/abjad.ily'],
3779         )
3780     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
3781     pdf_path = f'{directory}/Section_E.pdf'
3782     path = pathlib.Path('Section_E.pdf')
3783     if path.exists():
3784         print(f'Removing {pdf_path} ...')
3785         path.unlink()
3786     time_1 = time.time()
3787     print(f'Persisting {pdf_path} ...')
3788     result = abjad.persist(part_file).as_pdf(pdf_path)
3789     print(result[0])
3790     print(result[1])
3791     print(result[2])
3792     success = result[3]
3793     if success is False:
3794         print('LilyPond failed!')
3795     time_2 = time.time()
3796     total_time = time_2 - time_1
3797     print(f'Total time: {total_time} seconds')
3798     if path.exists():
3799         print(f'Opening {pdf_path} ...')
3800         os.system(f'open {pdf_path}')
3801     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
3802 Section_E.ly').readlines()
3803     open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_E.ly', 'w'
3804     ).writelines(part_lines[15:-1])

3805
3806 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
3807     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3808     signature_copy = abjad.mutate(signatures).copy()
3809     staff_copy = abjad.mutate(staff).copy()
3810     part = abjad.Score()
3811     part.insert(0, staff)
3812     part.insert(0, signature_copy)
3813     part_file = abjad.LilyPondFile.new(
3814         part,
3815         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3816 _stylesheets/abjad.ily'],
3817         )
3818     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
3819     pdf_path = f'{directory}/Section_E.pdf'
3820     path = pathlib.Path('Section_E.pdf')
3821     if path.exists():
3822         print(f'Removing {pdf_path} ...')
3823         path.unlink()
3824     time_1 = time.time()
3825     print(f'Persisting {pdf_path} ...')

```

```

3822     result = abjad.persist(part_file).as_pdf(pdf_path)
3823     print(result[0])
3824     print(result[1])
3825     print(result[2])
3826     success = result[3]
3827     if success is False:
3828         print('LilyPond failed!')
3829     time_2 = time.time()
3830     total_time = time_2 - time_1
3831     print(f'Total time: {total_time} seconds')
3832     if path.exists():
3833         print(f'Opening {pdf_path} ...')
3834         os.system(f'open {pdf_path}')
3835     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.')tenor1/
3836     Section_E.ly').readlines()
3837     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.')tenor1/Section_E.ly', ,
3838     w').writelines(part_lines[15:-1])
3839
3840 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
3841     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3842     signature_copy = abjad.mutate(signatures).copy()
3843     staff_copy = abjad.mutate(staff).copy()
3844     part = abjad.Score()
3845     part.insert(0, staff)
3846     part.insert(0, signature_copy)
3847     part_file = abjad.LilyPondFile.new(
3848         part,
3849         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3850         _stylesheets/abjad.ily'],
3851         )
3852     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2'
3853     pdf_path = f'{directory}/Section_E.pdf'
3854     path = pathlib.Path('Section_E.pdf')
3855     if path.exists():
3856         print(f'Removing {pdf_path} ...')
3857         path.unlink()
3858     time_1 = time.time()
3859     print(f'Persisting {pdf_path} ...')
3860     result = abjad.persist(part_file).as_pdf(pdf_path)
3861     print(result[0])
3862     print(result[1])
3863     print(result[2])
3864     success = result[3]
3865     if success is False:
3866         print('LilyPond failed!')
3867     time_2 = time.time()
3868     total_time = time_2 - time_1
3869     print(f'Total time: {total_time} seconds')
3870     if path.exists():
3871         print(f'Opening {pdf_path} ...')
3872         os.system(f'open {pdf_path}')
3873     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/
3874     Section_E.ly').readlines()
3875     open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/Section_E.ly', ,
3876     w').writelines(part_lines[15:-1])
3877
3878 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
3879     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3880     signature_copy = abjad.mutate(signatures).copy()

```

```

3876 staff_copy = abjad.mutate(staff).copy()
3877 part = abjad.Score()
3878 part.insert(0, staff)
3879 part.insert(0, signature_copy)
3880 part_file = abjad.LilyPondFile.new(
3881     part,
3882     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3883 _stylesheets/abjad.ily'],
3884     )
3885 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3'
3886 pdf_path = f'{directory}/Section_E.pdf'
3887 path = pathlib.Path('Section_E.pdf')
3888 if path.exists():
3889     print(f'Removing {pdf_path} ...')
3890     path.unlink()
3891 time_1 = time.time()
3892 print(f'Persisting {pdf_path} ...')
3893 result = abjad.persist(part_file).as_pdf(pdf_path)
3894 print(result[0])
3895 print(result[1])
3896 print(result[2])
3897 success = result[3]
3898 if success is False:
3899     print('LilyPond failed!')
3900 time_2 = time.time()
3901 total_time = time_2 - time_1
3902 print(f'Total time: {total_time} seconds')
3903 if path.exists():
3904     print(f'Opening {pdf_path} ...')
3905     os.system(f'open {pdf_path}')
3906 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
3907 Section_E.ly').readlines()
3908 open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_E.ly', ,
3909 w').writelines(part_lines[15:-1])
3910
3911 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
3912     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3913     signature_copy = abjad.mutate(signatures).copy()
3914     staff_copy = abjad.mutate(staff).copy()
3915     part = abjad.Score()
3916     part.insert(0, staff)
3917     part.insert(0, signature_copy)
3918     part_file = abjad.LilyPondFile.new(
3919         part,
3920         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3921 _stylesheets/abjad.ily'],
3922         )
3923     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'
3924     pdf_path = f'{directory}/Section_E.pdf'
3925     path = pathlib.Path('Section_E.pdf')
3926     if path.exists():
3927         print(f'Removing {pdf_path} ...')
3928         path.unlink()
3929     time_1 = time.time()
3930     print(f'Persisting {pdf_path} ...')
3931     result = abjad.persist(part_file).as_pdf(pdf_path)
3932     print(result[0])
3933     print(result[1])
3934     print(result[2])

```

```

3931     success = result[3]
3932     if success is False:
3933         print('LilyPond failed!')
3934     time_2 = time.time()
3935     total_time = time_2 - time_1
3936     print(f'Total time: {total_time} seconds')
3937     if path.exists():
3938         print(f'Opening {pdf_path} ...')
3939         os.system(f'open {pdf_path}')
3940     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
3941     Section_E.ly').readlines()
3942     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_E.ly', ,
3943     w').writelines(part_lines[15:-1])
3944
3945 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
3946     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3947     signature_copy = abjad.mutate(signatures).copy()
3948     staff_copy = abjad.mutate(staff).copy()
3949     part = abjad.Score()
3950     part.insert(0, staff)
3951     part.insert(0, signature_copy)
3952     part_file = abjad.LilyPondFile.new(
3953         part,
3954         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3955         _stylesheets/abjad.ily'],
3956         )
3957     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5'
3958     pdf_path = f'{directory}/Section_E.pdf'
3959     path = pathlib.Path('Section_E.pdf')
3960     if path.exists():
3961         print(f'Removing {pdf_path} ...')
3962         path.unlink()
3963     time_1 = time.time()
3964     print(f'Persisting {pdf_path} ...')
3965     result = abjad.persist(part_file).as_pdf(pdf_path)
3966     print(result[0])
3967     print(result[1])
3968     print(result[2])
3969     success = result[3]
3970     if success is False:
3971         print('LilyPond failed!')
3972     time_2 = time.time()
3973     total_time = time_2 - time_1
3974     print(f'Total time: {total_time} seconds')
3975     if path.exists():
3976         print(f'Opening {pdf_path} ...')
3977         os.system(f'open {pdf_path}')
3978     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/
3979     Section_E.ly').readlines()
3980     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/Section_E.ly', ,
3981     w').writelines(part_lines[15:-1])
3982
3983 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
3984     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3985     signature_copy = abjad.mutate(signatures).copy()
3986     staff_copy = abjad.mutate(staff).copy()
3987     part = abjad.Score()
3988     part.insert(0, staff)
3989     part.insert(0, signature_copy)

```

```

3985     part_file = abjad.LilyPondFile.new(
3986         part,
3987         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3988 _stylesheets/abjad.ily'],
3989         )
3990     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1'
3991     pdf_path = f'{directory}/Section_E.pdf'
3992     path = pathlib.Path('Section_E.pdf')
3993     if path.exists():
3994         print(f'Removing {pdf_path} ...')
3995         path.unlink()
3996     time_1 = time.time()
3997     print(f'Persisting {pdf_path} ...')
3998     result = abjad.persist(part_file).as_pdf(pdf_path)
3999     print(result[0])
4000     print(result[1])
4001     print(result[2])
4002     success = result[3]
4003     if success is False:
4004         print('LilyPond failed!')
4005     time_2 = time.time()
4006     total_time = time_2 - time_1
4007     print(f'Total time: {total_time} seconds')
4008     if path.exists():
4009         print(f'Opening {pdf_path} ...')
4010         os.system(f'open {pdf_path}')
4011     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/
4012 Section_E.ly').readlines()
4013     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/Section_E.ly',
4014         'w').writelines(part_lines[15:-1])
4015
4016 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
4017     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4018     signature_copy = abjad.mutate(signatures).copy()
4019     staff_copy = abjad.mutate(staff).copy()
4020     part = abjad.Score()
4021     part.insert(0, staff)
4022     part.insert(0, signature_copy)
4023     part_file = abjad.LilyPondFile.new(
4024         part,
4025         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
4026 _stylesheets/abjad.ily'],
4027         )
4028     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2'
4029     pdf_path = f'{directory}/Section_E.pdf'
4030     path = pathlib.Path('Section_E.pdf')
4031     if path.exists():
4032         print(f'Removing {pdf_path} ...')
4033         path.unlink()
4034     time_1 = time.time()
4035     print(f'Persisting {pdf_path} ...')
4036     result = abjad.persist(part_file).as_pdf(pdf_path)
4037     print(result[0])
4038     print(result[1])
4039     print(result[2])
4040     success = result[3]
4041     if success is False:
4042         print('LilyPond failed!')
4043     time_2 = time.time()

```

```

4040     total_time = time_2 - time_1
4041     print(f'Total time: {total_time} seconds')
4042     if path.exists():
4043         print(f'Opening {pdf_path} ...')
4044         os.system(f'open {pdf_path}')
4045     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/
4046     Section_E.ly').readlines()
4047     open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/Section_E.ly',
4048     , 'w').writelines(part_lines[15:-1])
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092

```

```

4093     )
4094     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
4095     pdf_path = f'{directory}/Section_E.pdf'
4096     path = pathlib.Path('Section_E.pdf')
4097     if path.exists():
4098         print(f'Removing {pdf_path} ...')
4099         path.unlink()
4100     time_1 = time.time()
4101     print(f'Persisting {pdf_path} ...')
4102     result = abjad.persist(part_file).as_pdf(pdf_path)
4103     print(result[0])
4104     print(result[1])
4105     print(result[2])
4106     success = result[3]
4107     if success is False:
4108         print('LilyPond failed!')
4109     time_2 = time.time()
4110     total_time = time_2 - time_1
4111     print(f'Total time: {total_time} seconds')
4112     if path.exists():
4113         print(f'Opening {pdf_path} ...')
4114         os.system(f'open {pdf_path}')
4115     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
4116     Section_E.ly').readlines()
4116     open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_E.ly', 'w'
4117     ).writelines(part_lines[15:-1])

4118 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
4119     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4120     signature_copy = abjad.mutate(signatures).copy()
4121     staff_copy = abjad.mutate(staff).copy()
4122     part = abjad.Score()
4123     part.insert(0, staff)
4124     part.insert(0, signature_copy)
4125     part_file = abjad.LilyPondFile.new(
4126         part,
4127         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
4128         _stylesheets/abjad.ly'],
4129         )
4129     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
4130     pdf_path = f'{directory}/Section_E.pdf'
4131     path = pathlib.Path('Section_E.pdf')
4132     if path.exists():
4133         print(f'Removing {pdf_path} ...')
4134         path.unlink()
4135     time_1 = time.time()
4136     print(f'Persisting {pdf_path} ...')
4137     result = abjad.persist(part_file).as_pdf(pdf_path)
4138     print(result[0])
4139     print(result[1])
4140     print(result[2])
4141     success = result[3]
4142     if success is False:
4143         print('LilyPond failed!')
4144     time_2 = time.time()
4145     total_time = time_2 - time_1
4146     print(f'Total time: {total_time} seconds')
4147     if path.exists():
4148         print(f'Opening {pdf_path} ...')

```

```

4149     os.system(f'open {pdf_path}')
4150     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
4151     Section_E.ly').readlines()
4152     open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_E.ly', 'w'
4153     ).writelines(part_lines[15:-1])
4154
4155 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
4156     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4157     signature_copy = abjad.mutate(signatures).copy()
4158     staff_copy = abjad.mutate(staff).copy()
4159     part = abjad.Score()
4160     part.insert(0, staff)
4161     part.insert(0, signature_copy)
4162     part_file = abjad.LilyPondFile.new(
4163         part,
4164         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
4165         _stylesheets/abjad.ily'],
4166         )
4167     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
4168     pdf_path = f'{directory}/Section_E.pdf'
4169     path = pathlib.Path('Section_E.pdf')
4170     if path.exists():
4171         print(f'Removing {pdf_path} ...')
4172         path.unlink()
4173     time_1 = time.time()
4174     print(f'Persisting {pdf_path} ...')
4175     result = abjad.persist(part_file).as_pdf(pdf_path)
4176     print(result[0])
4177     print(result[1])
4178     print(result[2])
4179     success = result[3]
4180     if success is False:
4181         print('LilyPond failed!')
4182     time_2 = time.time()
4183     total_time = time_2 - time_1
4184     print(f'Total time: {total_time} seconds')
4185     if path.exists():
4186         print(f'Opening {pdf_path} ...')
4187         os.system(f'open {pdf_path}')
4188         part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass
4189         /Section_E.ly').readlines()
4190         open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_E.ly
4191         ', 'w').writelines(part_lines[15:-1])

```

Listing 3.6: Invocation Source Code

### 3.7 Section F

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random

```

```

10 from random import seed
11 from TrillHandler import TrillHandler
12
13 print('Interpreting file ...')
14
15 time_signatures = [
16     abjad.TimeSignature(pair) for pair in [
17         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
18         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
19         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
20         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
21         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
22     ]
23 ]
24
25 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
26
27 def cyc(lst):
28     count = 0
29     while True:
30         yield lst[count%len(lst)]
31         count += 1
32
33 def grouper(lst1, lst2):
34     def cyc(lst):
35         c = 0
36         while True:
37             yield lst[c%len(lst)]
38             c += 1
39     lst1 = cyc(lst1)
40     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2
41 ]
42
43 def reduceMod(list_length, rw):
44     return [(x % list_length) for x in rw]
45
46 # -3 at bottom of chord for completion
47 soprano_note = [27.75, 11.5, 17.25, 8.5, 0.75, ]
48 soprano_1_note = [13.25, 16.5, 22.5, 5.25, ]
49 soprano_2_note = [13.5, 16.75, 22.25, 5.75, 16.5, ]
50 soprano_3_note = [13.25, 16.5, 22.5, 5.25, ]
51 alto_1_note = [23.75, 20.5, 12.25, 0.5, ]
52 alto_2_note = [23.5, 20.25, 12.5, 0.75, ]
53 alto_3_note = [23.25, 20.5, 12.75, 0.5, ]
54 alto_4_note = [23.5, 20.75, 12.5, 0.25, ]
55 alto_5_note = [23.75, 20.5, 12.25, 0.5, ]
56 alto_6_note = [23.5, 20.25, 12.5, 0.75, ]
57 tenor_1_note = [25.5, 6.25, 17.5, ]
58 tenor_2_note = [25.25, 6.5, 17.75, ]
59 tenor_3_note = [25.5, 6.75, 17.5, ]
60 tenor_4_note = [25.75, 6.5, 17.25, ]
61 tenor_5_note = [25.5, 6.25, 17.5, ]
62 baritone_1_note = [13.25, 24.5, 4.75, 6.5, ]
63 baritone_2_note = [13.25, 24.5, 4.75, 6.5, ]
64 baritone_3_note = [13.25, 24.5, 4.75, 6.5, ]
65 bass_1_note = [11.25, 18.5, 9.75, 0.5, ]
66 bass_2_note = [11.25, 18.5, 9.75, 0.5, ]
67 contrabass_note = [2.25, -2.5, 7.75, 18.5, 16.25, 25.5, ]

```

```

68 soprano_trill = [[17, 27, ], [8, 11, ], [0, 8, ], [17, 11, ]]
69 soprano_1_trill = [[5, 13, ], [22, 16, ], [16, 5, ]]
70 soprano_2_trill = [[22, 16, ], [5, 13, ], [16, 5, ]]
71 soprano_3_trill = [[5, 13, ], [22, 16, ], [16, 5, ]]
72 alto_1_trill = [[23, 20, ], [1, 12, ], [12, 20, ]]
73 alto_2_trill = [[23, 20, ], [12, 20], [1, 12, ], [12, 23, ]]
74 alto_3_trill = [[1, 12, ], [23, 20, ], [12, 20, ]]
75 alto_4_trill = [[12, 20], [1, 12, ], [23, 20, ], [12, 23, ]]
76 alto_5_trill = [[1, 12, ], [23, 20, ], [12, 20]]
77 alto_6_trill = [[23, 20, ], [12, 23, ], [1, 12, ], [12, 20]]
78 tenor_1_trill = [[-1, 6, ], [17, 25, ], [6, 17, ], ]
79 tenor_2_trill = [[6, 17, ], [-1, 6, ], [17, 25, ]]
80 tenor_3_trill = [[6, 17, ], [17, 25, ], [-1, 6, ]]
81 tenor_4_trill = [[6, 17, ], [17, 25, ], [-1, 6, ]]
82 tenor_5_trill = [[-1, 6, ], [6, 17, ], [17, 25, ]]
83 baritone_1_trill = [[4, 6, ], [24, 13, ], [6, 13, ]]
84 baritone_2_trill = [[4, 6, ], [6, 13, ], [24, 13, ]]
85 baritone_3_trill = [[24, 13, ], [6, 13, ], [4, 6, ]]
86 bass_1_trill = [[0, 9, ], [18, 11, ], [11, 9, ]]
87 bass_2_trill = [[18, 11, ], [0, 9, ], [11, 9, ]]
88 contrabass_trill = [[-2, 2, ], [25, 18, ], [7, 16, ], [7, 18, ], ]
89
90 def reduceMod(x, rw):
91     return [(y % x) for y in rw]
92
93 seed(1)
94 soprano_random_walk = []
95 soprano_random_walk.append(-1 if random() < 0.5 else 1)
96 for i in range(1, 1000):
97     movement = -1 if random() < 0.5 else 1
98     value = soprano_random_walk[i-1] + movement
99     soprano_random_walk.append(value)
100    soprano_walk_chord = [11, 27, 17, 0, 8, ]
101 l = len(soprano_walk_chord)
102 soprano_random_walk_notes = [soprano_walk_chord[x] for x in reduceMod(l,
103                         soprano_random_walk)]
104
104 seed(2)
105 soprano_1_random_walk = []
106 soprano_1_random_walk.append(-1 if random() < 0.5 else 1)
107 for i in range(1, 1000):
108     movement = -1 if random() < 0.5 else 1
109     value = soprano_1_random_walk[i-1] + movement
110     soprano_1_random_walk.append(value)
111    soprano_1_walk_chord = [13, 5, 16, 22, ]
112 l = len(soprano_1_walk_chord)
113 soprano_1_random_walk_notes = [soprano_1_walk_chord[x] for x in reduceMod(l,
114                         soprano_1_random_walk)]
114
115 seed(3)
116 soprano_2_random_walk = []
117 soprano_2_random_walk.append(-1 if random() < 0.5 else 1)
118 for i in range(1, 1000):
119     movement = -1 if random() < 0.5 else 1
120     value = soprano_2_random_walk[i-1] + movement
121     soprano_2_random_walk.append(value)
122    soprano_2_random_walk.append(value)
123    soprano_2_walk_chord = [16, 22, 13, 5, ]
124 l = len(soprano_2_walk_chord)

```

```
125 soprano_2_random_walk_notes = [soprano_2_walk_chord[x] for x in reduceMod(1,
126   soprano_2_random_walk)]
127 seed(4)
128 soprano_3_random_walk = []
129 soprano_3_random_walk.append(-1 if random() < 0.5 else 1)
130 for i in range(1, 1000):
131   movement = -1 if random() < 0.5 else 1
132   value = soprano_3_random_walk[i-1] + movement
133   soprano_3_random_walk.append(value)
134 soprano_3_random_walk.append(value)
135 soprano_3_walk_chord = [16, 5, 22, 13, ]
136 l = len(soprano_3_walk_chord)
137 soprano_3_random_walk_notes = [soprano_3_walk_chord[x] for x in reduceMod(l,
138   soprano_3_random_walk)]
139 seed(5)
140 alto_1_random_walk = []
141 alto_1_random_walk.append(-1 if random() < 0.5 else 1)
142 for i in range(1, 1000):
143   movement = -1 if random() < 0.5 else 1
144   value = alto_1_random_walk[i-1] + movement
145   alto_1_random_walk.append(value)
146 alto_1_walk_chord = [12, 23, 20, 1, 12, 20, ]
147 l = len(alto_1_walk_chord)
148 alto_1_random_walk_notes = [alto_1_walk_chord[x] for x in reduceMod(l,
149   alto_1_random_walk)]
150 seed(6)
151 alto_2_random_walk = []
152 alto_2_random_walk.append(-1 if random() < 0.5 else 1)
153 for i in range(1, 1000):
154   movement = -1 if random() < 0.5 else 1
155   value = alto_2_random_walk[i-1] + movement
156   alto_2_random_walk.append(value)
157 alto_2_walk_chord = [23, 20, 12, 23, 1, 12, 20, ]
158 l = len(alto_2_walk_chord)
159 alto_2_random_walk_notes = [alto_2_walk_chord[x] for x in reduceMod(l,
160   alto_2_random_walk)]
161 seed(7)
162 alto_3_random_walk = []
163 alto_3_random_walk.append(-1 if random() < 0.5 else 1)
164 for i in range(1, 1000):
165   movement = -1 if random() < 0.5 else 1
166   value = alto_3_random_walk[i-1] + movement
167   alto_3_random_walk.append(value)
168 alto_3_walk_chord = [23, 20, 12, 1, 12, 20, ]
169 l = len(alto_3_walk_chord)
170 alto_3_random_walk_notes = [alto_3_walk_chord[x] for x in reduceMod(l,
171   alto_3_random_walk)]
172 seed(8)
173 alto_4_random_walk = []
174 alto_4_random_walk.append(-1 if random() < 0.5 else 1)
175 for i in range(1, 1000):
176   movement = -1 if random() < 0.5 else 1
177   value = alto_4_random_walk[i-1] + movement
178   alto_4_random_walk.append(value)
```

```

179 alto_4_walk_chord = [23, 1, 12, 20, 23, 20, 12, ]
180 l = len(alto_4_walk_chord)
181 alto_4_random_walk_notes = [alto_4_walk_chord[x] for x in reduceMod(l,
182                         alto_4_random_walk)]
183
184 seed(9)
185 alto_5_random_walk = []
186 alto_5_random_walk.append(-1 if random() < 0.5 else 1)
187 for i in range(1, 1000):
188     movement = -1 if random() < 0.5 else 1
189     value = alto_5_random_walk[i-1] + movement
190     alto_5_random_walk.append(value)
191 alto_5_walk_chord = [23, 1, 12, 20, 23, 20, 12, ]
192 l = len(alto_5_walk_chord)
193 alto_5_random_walk_notes = [alto_5_walk_chord[x] for x in reduceMod(l,
194                         alto_5_random_walk)]
195
196 seed(10)
197 alto_6_random_walk = []
198 alto_6_random_walk.append(-1 if random() < 0.5 else 1)
199 for i in range(1, 1000):
200     movement = -1 if random() < 0.5 else 1
201     value = alto_6_random_walk[i-1] + movement
202     alto_6_random_walk.append(value)
203 alto_6_walk_chord = [23, 20, 12, 1, 12, 20, 23, ]
204 l = len(alto_6_walk_chord)
205 alto_6_random_walk_notes = [alto_6_walk_chord[x] for x in reduceMod(l,
206                         alto_6_random_walk)]
207
208 seed(11)
209 tenor_1_random_walk = []
210 tenor_1_random_walk.append(-1 if random() < 0.5 else 1)
211 for i in range(1, 1000):
212     movement = -1 if random() < 0.5 else 1
213     value = tenor_1_random_walk[i-1] + movement
214     tenor_1_random_walk.append(value)
215 tenor_1_walk_chord = [-1, 17, 25, 17, 6, ]
216 l = len(tenor_1_walk_chord)
217 tenor_1_random_walk_notes = [tenor_1_walk_chord[x] for x in reduceMod(l,
218                         tenor_1_random_walk)]
219
220 seed(12)
221 tenor_2_random_walk = []
222 tenor_2_random_walk.append(-1 if random() < 0.5 else 1)
223 for i in range(1, 1000):
224     movement = -1 if random() < 0.5 else 1
225     value = tenor_2_random_walk[i-1] + movement
226     tenor_2_random_walk.append(value)
227 tenor_2_walk_chord = [-1, 17, 25, 17, 6, ]
228 l = len(tenor_2_walk_chord)
229 tenor_2_random_walk_notes = [tenor_2_walk_chord[x] for x in reduceMod(l,
230                         tenor_2_random_walk)]
231
232 seed(13)
233 tenor_3_random_walk = []
234 tenor_3_random_walk.append(-1 if random() < 0.5 else 1)
235 for i in range(1, 1000):
236     movement = -1 if random() < 0.5 else 1
237     value = tenor_3_random_walk[i-1] + movement

```

```

233     tenor_3_random_walk.append(value)
234 tenor_3_walk_chord = [17, 6, -1, 17, 25, ]
235 l = len(tenor_3_walk_chord)
236 tenor_3_random_walk_notes = [tenor_3_walk_chord[x] for x in reduceMod(1,
237                           tenor_3_random_walk)]
238
239 seed(14)
240 tenor_4_random_walk = []
241 tenor_4_random_walk.append(-1 if random() < 0.5 else 1)
242 for i in range(1, 1000):
243     movement = -1 if random() < 0.5 else 1
244     value = tenor_4_random_walk[i-1] + movement
245     tenor_4_random_walk.append(value)
246 tenor_4_walk_chord = [17, 6, -1, 17, 25, ]
247 l = len(tenor_4_walk_chord)
248 tenor_4_random_walk_notes = [tenor_4_walk_chord[x] for x in reduceMod(1,
249                           tenor_4_random_walk)]
250
251 seed(15)
252 tenor_5_random_walk = []
253 tenor_5_random_walk.append(-1 if random() < 0.5 else 1)
254 for i in range(1, 1000):
255     movement = -1 if random() < 0.5 else 1
256     value = tenor_5_random_walk[i-1] + movement
257     tenor_5_random_walk.append(value)
258 tenor_5_walk_chord = [25, 17, 6, -1, 17, ]
259 l = len(tenor_5_walk_chord)
260 tenor_5_random_walk_notes = [tenor_5_walk_chord[x] for x in reduceMod(1,
261                           tenor_5_random_walk)]
262
263 seed(16)
264 baritone_1_random_walk = []
265 baritone_1_random_walk.append(-1 if random() < 0.5 else 1)
266 for i in range(1, 1000):
267     movement = -1 if random() < 0.5 else 1
268     value = baritone_1_random_walk[i-1] + movement
269     baritone_1_random_walk.append(value)
270 baritone_1_walk_chord = [6, 4, 13, 24, 13, ]
271 l = len(baritone_1_walk_chord)
272 baritone_1_random_walk_notes = [baritone_1_walk_chord[x] for x in reduceMod(1,
273                           baritone_1_random_walk)]
274
275 seed(17)
276 baritone_2_random_walk = []
277 baritone_2_random_walk.append(-1 if random() < 0.5 else 1)
278 for i in range(1, 1000):
279     movement = -1 if random() < 0.5 else 1
280     value = baritone_2_random_walk[i-1] + movement
281     baritone_2_random_walk.append(value)
282 baritone_2_walk_chord = [6, 13, 4, 13, 24, ]
283 l = len(baritone_2_walk_chord)
284 baritone_2_random_walk_notes = [baritone_2_walk_chord[x] for x in reduceMod(1,
285                           baritone_2_random_walk)]
286
287 seed(18)
288 baritone_3_random_walk = []
289 baritone_3_random_walk.append(-1 if random() < 0.5 else 1)
290 for i in range(1, 1000):
291     movement = -1 if random() < 0.5 else 1

```

```

287     value = baritone_3_random_walk[i-1] + movement
288     baritone_3_random_walk.append(value)
289 baritone_3_walk_chord = [6, 13, 24, 13, 4, ]
290 l = len(baritone_3_walk_chord)
291 baritone_3_random_walk_notes = [baritone_3_walk_chord[x] for x in reduceMod(l,
292                             baritone_3_random_walk)]
293
294 seed(19)
295 bass_1_random_walk = []
296 bass_1_random_walk.append(-1 if random() < 0.5 else 1)
297 for i in range(1, 1000):
298     movement = -1 if random() < 0.5 else 1
299     value = bass_1_random_walk[i-1] + movement
300     bass_1_random_walk.append(value)
301 bass_1_walk_chord = [11, 9, 0, 18, ]
302 l = len(bass_1_walk_chord)
303 bass_1_random_walk_notes = [bass_1_walk_chord[x] for x in reduceMod(l,
304                             bass_1_random_walk)]
305
306 seed(20)
307 bass_2_random_walk = []
308 bass_2_random_walk.append(-1 if random() < 0.5 else 1)
309 for i in range(1, 1000):
310     movement = -1 if random() < 0.5 else 1
311     value = bass_2_random_walk[i-1] + movement
312     bass_2_random_walk.append(value)
313 bass_2_walk_chord = [0, 9, 18, 11, ]
314 l = len(bass_2_walk_chord)
315 bass_2_random_walk_notes = [bass_2_walk_chord[x] for x in reduceMod(l,
316                             bass_2_random_walk)]
317
318 seed(21)
319 contrabass_random_walk = []
320 contrabass_random_walk.append(-1 if random() < 0.5 else 1)
321 for i in range(1, 1000):
322     movement = -1 if random() < 0.5 else 1
323     value = contrabass_random_walk[i-1] + movement
324     contrabass_random_walk.append(value)
325 contrabass_walk_chord = [18, 7, 16, 2, -2, 16, 25, ]
326 l = len(contrabass_walk_chord)
327 contrabass_random_walk_notes = [contrabass_walk_chord[x] for x in reduceMod(l,
328                             contrabass_random_walk)]
329
330 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
331     talea=abjadext.rmakers.Talea(
332         counts=[11, 8, 12, 7, 10, 9, ],
333         denominator=16,
334         ),
335     beamSpecifier=abjadext.rmakers.BeamSpecifier(
336         beamDivisionsTogether=True,
337         beamRests=False,
338         ),
339     extraCountsPerDivision=[0, -1, 0, 0, 1, -1, 1, ],
340     logicalTieMasks=[
341         abjadext.rmakers.silence([8], 11),
342         ],
343     divisionMasks=[
344         abjadext.rmakers.SilenceMask(
345             pattern=abjad.index([7], 17),

```

```

342     ),
343     ],
344     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
345         trivialize=True,
346         extract_trivial=True,
347         rewrite_rest_filled=True,
348         rewrite_dots=True,
349         rewrite_sustained=True,
350         denominator='divisions',
351         ),
352     )
353
354 rmaker_two = abjadext.rmakers.EvenDivisionRhythmMaker(
355     denominators=[16, 16, 8, 16, 4, 8, 4, 16, 8, ],
356     extra_counts_per_division=[0, 1, -1, 0, 1, 0, -1, ],
357     logical_tie_masks=[
358         abjadext.rmakers.silence([2], 7),
359     ],
360     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
361         trivialize=True,
362         extract_trivial=True,
363         rewrite_rest_filled=True,
364         rewrite_sustained=True,
365         denominator='divisions',
366         ),
367     )
368
369 attachment_handler_one = AttachmentHandler(
370     starting_dynamic='pp',
371     ending_dynamic='f',
372     hairpin='<',
373     articulation_list=['tenuto', '', 'tenuto', 'halfopen', 'flageolet', 'halfopen',
374     '', 'tenuto', '', ],
375 )
376 attachment_handler_two = AttachmentHandler(
377     starting_dynamic='ff',
378     ending_dynamic='p',
379     hairpin='>',
380     articulation_list=['', 'portato', '', 'flageolet', 'halfopen', '', 'portato',
381     'flageolet', ],
382 )
383 attachment_handler_three = AttachmentHandler(
384     starting_dynamic='mf',
385     hairpin='--',
386 )
387
388 #####sopranino#####
389 sopranino_musicmaker_one = MusicMaker(
390     rmaker=rmaker_one,
391     pitches=sopranino_note,
392     continuous=True,
393     attachment_handler=attachment_handler_one,
394 )
395 sopranino_musicmaker_two = MusicMaker(
396     rmaker=rmaker_one,
397     pitches=sopranino_trill,
398     continuous=True,

```

```
399     attachment_handler=attachment_handler_two ,
400 )
401 soprano_musicmaker_three = MusicMaker(
402     rmaker=rmaker_two ,
403     pitches=soprano_random_walk_notes ,
404     continuous=True ,
405     attachment_handler=attachment_handler_three ,
406 )
407 #####soprano_one#####
408 soprano_one_musicmaker_one = MusicMaker(
409     rmaker=rmaker_one ,
410     pitches=soprano_1_note ,
411     continuous=True ,
412     attachment_handler=attachment_handler_one ,
413 )
414 soprano_one_musicmaker_two = MusicMaker(
415     rmaker=rmaker_one ,
416     pitches=soprano_1_trill ,
417     continuous=True ,
418     attachment_handler=attachment_handler_two ,
419 )
420 soprano_one_musicmaker_three = MusicMaker(
421     rmaker=rmaker_two ,
422     pitches=soprano_1_random_walk_notes ,
423     continuous=True ,
424     attachment_handler=attachment_handler_three ,
425 )
426 #####soprano_two#####
427 soprano_two_musicmaker_one = MusicMaker(
428     rmaker=rmaker_one ,
429     pitches=soprano_2_note ,
430     continuous=True ,
431     attachment_handler=attachment_handler_one ,
432 )
433 soprano_two_musicmaker_two = MusicMaker(
434     rmaker=rmaker_one ,
435     pitches=soprano_2_trill ,
436     continuous=True ,
437     attachment_handler=attachment_handler_two ,
438 )
439 soprano_two_musicmaker_three = MusicMaker(
440     rmaker=rmaker_two ,
441     pitches=soprano_2_random_walk_notes ,
442     continuous=True ,
443     attachment_handler=attachment_handler_three ,
444 )
445 #####soprano_three#####
446 soprano_three_musicmaker_one = MusicMaker(
447     rmaker=rmaker_one ,
448     pitches=soprano_3_note ,
449     continuous=True ,
450     attachment_handler=attachment_handler_one ,
451 )
452 soprano_three_musicmaker_two = MusicMaker(
453     rmaker=rmaker_one ,
454     pitches=soprano_3_trill ,
455     continuous=True ,
456     attachment_handler=attachment_handler_two ,
457 )
```

```
458 soprano_three_musicmaker_three = MusicMaker(
459     rmaker=rmaker_two,
460     pitches=soprano_3_random_walk_notes,
461     continuous=True,
462     attachment_handler=attachment_handler_three,
463 )
464 #####alto_one#####
465 alto_one_musicmaker_one = MusicMaker(
466     rmaker=rmaker_one,
467     pitches=alto_1_note,
468     continuous=True,
469     attachment_handler=attachment_handler_one,
470 )
471 alto_one_musicmaker_two = MusicMaker(
472     rmaker=rmaker_one,
473     pitches=soprano_1_trill,
474     continuous=True,
475     attachment_handler=attachment_handler_two,
476 )
477 alto_one_musicmaker_three = MusicMaker(
478     rmaker=rmaker_two,
479     pitches=soprano_1_random_walk_notes,
480     continuous=True,
481     attachment_handler=attachment_handler_three,
482 )
483 #####alto_two#####
484 alto_two_musicmaker_one = MusicMaker(
485     rmaker=rmaker_one,
486     pitches=alto_2_note,
487     continuous=True,
488     attachment_handler=attachment_handler_one,
489 )
490 alto_two_musicmaker_two = MusicMaker(
491     rmaker=rmaker_one,
492     pitches=soprano_2_trill,
493     continuous=True,
494     attachment_handler=attachment_handler_two,
495 )
496 alto_two_musicmaker_three = MusicMaker(
497     rmaker=rmaker_two,
498     pitches=soprano_2_random_walk_notes,
499     continuous=True,
500     attachment_handler=attachment_handler_three,
501 )
502 #####alto_three#####
503 alto_three_musicmaker_one = MusicMaker(
504     rmaker=rmaker_one,
505     pitches=alto_3_note,
506     continuous=True,
507     attachment_handler=attachment_handler_one,
508 )
509 alto_three_musicmaker_two = MusicMaker(
510     rmaker=rmaker_one,
511     pitches=soprano_3_trill,
512     continuous=True,
513     attachment_handler=attachment_handler_two,
514 )
515 alto_three_musicmaker_three = MusicMaker(
516     rmaker=rmaker_two,
```

```
517     pitches=soprano_3_random_walk_notes ,
518     continuous=True ,
519     attachment_handler=attachment_handler_three ,
520 )
521 #####alto_four#####
522 alto_four_musicmaker_one = MusicMaker(
523     rmaker=rmaker_one ,
524     pitches=alto_4_note ,
525     continuous=True ,
526     attachment_handler=attachment_handler_one ,
527 )
528 alto_four_musicmaker_two = MusicMaker(
529     rmaker=rmaker_one ,
530     pitches=alto_4_trill ,
531     continuous=True ,
532     attachment_handler=attachment_handler_two ,
533 )
534 alto_four_musicmaker_three = MusicMaker(
535     rmaker=rmaker_two ,
536     pitches=alto_4_random_walk_notes ,
537     continuous=True ,
538     attachment_handler=attachment_handler_three ,
539 )
540 #####alto_five#####
541 alto_five_musicmaker_one = MusicMaker(
542     rmaker=rmaker_one ,
543     pitches=alto_5_note ,
544     continuous=True ,
545     attachment_handler=attachment_handler_one ,
546 )
547 alto_five_musicmaker_two = MusicMaker(
548     rmaker=rmaker_one ,
549     pitches=alto_5_trill ,
550     continuous=True ,
551     attachment_handler=attachment_handler_two ,
552 )
553 alto_five_musicmaker_three = MusicMaker(
554     rmaker=rmaker_two ,
555     pitches=alto_5_random_walk_notes ,
556     continuous=True ,
557     attachment_handler=attachment_handler_three ,
558 )
559 #####alto_six#####
560 alto_six_musicmaker_one = MusicMaker(
561     rmaker=rmaker_one ,
562     pitches=alto_6_note ,
563     continuous=True ,
564     attachment_handler=attachment_handler_one ,
565 )
566 alto_six_musicmaker_two = MusicMaker(
567     rmaker=rmaker_one ,
568     pitches=alto_6_trill ,
569     continuous=True ,
570     attachment_handler=attachment_handler_two ,
571 )
572 alto_six_musicmaker_three = MusicMaker(
573     rmaker=rmaker_two ,
574     pitches=alto_6_random_walk_notes ,
575     continuous=True ,
```

```
576     attachment_handler=attachment_handler_three,
577 )
578 #####tenor_one#####
579 tenor_one_musicmaker_one = MusicMaker(
580     rmaker=rmaker_one,
581     pitches=tenor_1_note,
582     continuous=True,
583     attachment_handler=attachment_handler_one,
584 )
585 tenor_one_musicmaker_two = MusicMaker(
586     rmaker=rmaker_one,
587     pitches=tenor_1_trill,
588     continuous=True,
589     attachment_handler=attachment_handler_two,
590 )
591 tenor_one_musicmaker_three = MusicMaker(
592     rmaker=rmaker_two,
593     pitches=tenor_1_random_walk_notes,
594     continuous=True,
595     attachment_handler=attachment_handler_three,
596 )
597 #####tenor_two#####
598 tenor_two_musicmaker_one = MusicMaker(
599     rmaker=rmaker_one,
600     pitches=tenor_2_note,
601     continuous=True,
602     attachment_handler=attachment_handler_one,
603 )
604 tenor_two_musicmaker_two = MusicMaker(
605     rmaker=rmaker_one,
606     pitches=tenor_2_trill,
607     continuous=True,
608     attachment_handler=attachment_handler_two,
609 )
610 tenor_two_musicmaker_three = MusicMaker(
611     rmaker=rmaker_two,
612     pitches=tenor_2_random_walk_notes,
613     continuous=True,
614     attachment_handler=attachment_handler_three,
615 )
616 #####tenor_three#####
617 tenor_three_musicmaker_one = MusicMaker(
618     rmaker=rmaker_one,
619     pitches=tenor_3_note,
620     continuous=True,
621     attachment_handler=attachment_handler_one,
622 )
623 tenor_three_musicmaker_two = MusicMaker(
624     rmaker=rmaker_one,
625     pitches=tenor_3_trill,
626     continuous=True,
627     attachment_handler=attachment_handler_two,
628 )
629 tenor_three_musicmaker_three = MusicMaker(
630     rmaker=rmaker_two,
631     pitches=tenor_3_random_walk_notes,
632     continuous=True,
633     attachment_handler=attachment_handler_three,
634 )
```

```
635 #####tenor_four#####
636 tenor_four_musicmaker_one = MusicMaker(
637     rmaker=rmaker_one,
638     pitches=tenor_4_note,
639     continuous=True,
640     attachment_handler=attachment_handler_one,
641 )
642 tenor_four_musicmaker_two = MusicMaker(
643     rmaker=rmaker_one,
644     pitches=tenor_4_trill,
645     continuous=True,
646     attachment_handler=attachment_handler_two,
647 )
648 tenor_four_musicmaker_three = MusicMaker(
649     rmaker=rmaker_two,
650     pitches=tenor_4_random_walk_notes,
651     continuous=True,
652     attachment_handler=attachment_handler_three,
653 )
654 #####tenor_five#####
655 tenor_five_musicmaker_one = MusicMaker(
656     rmaker=rmaker_one,
657     pitches=tenor_5_note,
658     continuous=True,
659     attachment_handler=attachment_handler_one,
660 )
661 tenor_five_musicmaker_two = MusicMaker(
662     rmaker=rmaker_one,
663     pitches=tenor_5_trill,
664     continuous=True,
665     attachment_handler=attachment_handler_two,
666 )
667 tenor_five_musicmaker_three = MusicMaker(
668     rmaker=rmaker_two,
669     pitches=tenor_5_random_walk_notes,
670     continuous=True,
671     attachment_handler=attachment_handler_three,
672 )
673 #####baritone_one#####
674 baritone_one_musicmaker_one = MusicMaker(
675     rmaker=rmaker_one,
676     pitches=baritone_1_note,
677     continuous=True,
678     attachment_handler=attachment_handler_one,
679 )
680 baritone_one_musicmaker_two = MusicMaker(
681     rmaker=rmaker_one,
682     pitches=baritone_1_trill,
683     continuous=True,
684     attachment_handler=attachment_handler_two,
685 )
686 baritone_one_musicmaker_three = MusicMaker(
687     rmaker=rmaker_two,
688     pitches=baritone_1_random_walk_notes,
689     continuous=True,
690     attachment_handler=attachment_handler_three,
691 )
692 #####baritone_two#####
693 baritone_two_musicmaker_one = MusicMaker(
```

```
694     rmaker=rmaker_one,
695     pitches=baritone_2_note,
696     continuous=True,
697     attachment_handler=attachment_handler_one,
698 )
699 baritone_two_musicmaker_two = MusicMaker(
700     rmaker=rmaker_one,
701     pitches=baritone_2_trill,
702     continuous=True,
703     attachment_handler=attachment_handler_two,
704 )
705 baritone_two_musicmaker_three = MusicMaker(
706     rmaker=rmaker_two,
707     pitches=baritone_2_random_walk_notes,
708     continuous=True,
709     attachment_handler=attachment_handler_three,
710 )
711 #####baritone_three#####
712 baritone_three_musicmaker_one = MusicMaker(
713     rmaker=rmaker_one,
714     pitches=baritone_3_note,
715     continuous=True,
716     attachment_handler=attachment_handler_one,
717 )
718 baritone_three_musicmaker_two = MusicMaker(
719     rmaker=rmaker_one,
720     pitches=baritone_3_trill,
721     continuous=True,
722     attachment_handler=attachment_handler_two,
723 )
724 baritone_three_musicmaker_three = MusicMaker(
725     rmaker=rmaker_two,
726     pitches=baritone_3_random_walk_notes,
727     continuous=True,
728     attachment_handler=attachment_handler_three,
729 )
730 #####bass_one#####
731 bass_one_musicmaker_one = MusicMaker(
732     rmaker=rmaker_one,
733     pitches=bass_1_note,
734     continuous=True,
735     attachment_handler=attachment_handler_one,
736 )
737 bass_one_musicmaker_two = MusicMaker(
738     rmaker=rmaker_one,
739     pitches=bass_1_trill,
740     continuous=True,
741     attachment_handler=attachment_handler_two,
742 )
743 bass_one_musicmaker_three = MusicMaker(
744     rmaker=rmaker_two,
745     pitches=bass_1_random_walk_notes,
746     continuous=True,
747     attachment_handler=attachment_handler_three,
748 )
749 #####bass_two#####
750 bass_two_musicmaker_one = MusicMaker(
751     rmaker=rmaker_one,
752     pitches=bass_2_note,
```

```

753     continuous=True,
754     attachment_handler=attachment_handler_one,
755 )
756 bass_two_musicmaker_two = MusicMaker(
757     rmaker=rmaker_one,
758     pitches=bass_2_trill,
759     continuous=True,
760     attachment_handler=attachment_handler_two,
761 )
762 bass_two_musicmaker_three = MusicMaker(
763     rmaker=rmaker_two,
764     pitches=bass_2_random_walk_notes,
765     continuous=True,
766     attachment_handler=attachment_handler_three,
767 )
768 #####contrabass#####
769 contrabass_musicmaker_one = MusicMaker(
770     rmaker=rmaker_one,
771     pitches=contrabass_note,
772     continuous=True,
773     attachment_handler=attachment_handler_one,
774 )
775 contrabass_musicmaker_two = MusicMaker(
776     rmaker=rmaker_one,
777     pitches=contrabass_trill,
778     continuous=True,
779     attachment_handler=attachment_handler_two,
780 )
781 contrabass_musicmaker_three = MusicMaker(
782     rmaker=rmaker_two,
783     pitches=contrabass_random_walk_notes,
784     continuous=True,
785     attachment_handler=attachment_handler_three,
786 )
787
788 silence_maker = abjadext.rmakers.NoteRhythmMaker(
789     division_masks=[
790         abjadext.rmakers.SilenceMask(
791             pattern=abjad.index([0], 1),
792             ),
793         ],
794     )
795
796 class MusicSpecifier:
797
798     def __init__(self, music_maker, voice_name):
799         self.music_maker = music_maker
800         self.voice_name = voice_name
801
802     print('Collecting timespans and rmakers ...')
803
804 voice_1_timespan_list = abjad.TimespanList([
805     abjad.AnnotatedTimespan(
806         start_offset=start_offset,
807         stop_offset=stop_offset,
808         annotation=MusicSpecifier(
809             music_maker=music_maker,
810             voice_name='Voice 1',
811         ),

```

```
812     )
813     for start_offset, stop_offset, music_maker in [
814         [(0, 8), (2, 8), sopranino_musicmaker_one],
815         [(2, 8), (4, 8), sopranino_musicmaker_one],
816         [(4, 8), (6, 8), sopranino_musicmaker_two],
817         [(6, 8), (8, 8), sopranino_musicmaker_one],
818         [(8, 8), (10, 8), sopranino_musicmaker_two],
819         [(10, 8), (12, 8), sopranino_musicmaker_two],
820         [(12, 8), (14, 8), sopranino_musicmaker_one],
821         [(14, 8), (16, 8), sopranino_musicmaker_one],
822         [(16, 8), (18, 8), sopranino_musicmaker_one],
823         [(18, 8), (20, 8), sopranino_musicmaker_two],
824         [(20, 8), (22, 8), sopranino_musicmaker_one],
825         [(22, 8), (24, 8), sopranino_musicmaker_one],
826         [(24, 8), (26, 8), sopranino_musicmaker_two],
827         [(26, 8), (28, 8), sopranino_musicmaker_one],
828         [(28, 8), (30, 8), sopranino_musicmaker_one],
829         [(30, 8), (32, 8), sopranino_musicmaker_two],
830         [(32, 8), (34, 8), sopranino_musicmaker_two],
831         [(34, 8), (36, 8), sopranino_musicmaker_one],
832         [(36, 8), (38, 8), sopranino_musicmaker_one],
833         [(38, 8), (40, 8), sopranino_musicmaker_one],
834         [(40, 8), (42, 8), sopranino_musicmaker_two],
835         [(42, 8), (44, 8), sopranino_musicmaker_one],
836         [(44, 8), (46, 8), sopranino_musicmaker_one],
837         [(46, 8), (48, 8), sopranino_musicmaker_one],
838         [(48, 8), (50, 8), sopranino_musicmaker_two],
839         [(50, 8), (52, 8), sopranino_musicmaker_two],
840         [(52, 8), (54, 8), sopranino_musicmaker_two],
841         [(54, 8), (56, 8), sopranino_musicmaker_one],
842         [(56, 8), (58, 8), sopranino_musicmaker_one],
843         [(58, 8), (60, 8), sopranino_musicmaker_two],
844         [(60, 8), (62, 8), sopranino_musicmaker_one],
845         [(62, 8), (64, 8), sopranino_musicmaker_two],
846         [(64, 8), (66, 8), sopranino_musicmaker_two],
847         [(66, 8), (68, 8), sopranino_musicmaker_one],
848         [(68, 8), (70, 8), sopranino_musicmaker_one],
849         [(70, 8), (72, 8), sopranino_musicmaker_one],
850         [(72, 8), (74, 8), sopranino_musicmaker_one],
851         [(74, 8), (76, 8), sopranino_musicmaker_one],
852         [(76, 8), (78, 8), sopranino_musicmaker_two],
853         [(78, 8), (80, 8), sopranino_musicmaker_two],
854         [(80, 8), (82, 8), sopranino_musicmaker_one],
855         [(82, 8), (84, 8), sopranino_musicmaker_one],
856         [(84, 8), (86, 8), sopranino_musicmaker_two],
857         [(86, 8), (88, 8), sopranino_musicmaker_one],
858         [(88, 8), (90, 8), sopranino_musicmaker_one],
859         [(90, 8), (92, 8), sopranino_musicmaker_two],
860         [(92, 8), (94, 8), sopranino_musicmaker_two],
861         [(94, 8), (96, 8), sopranino_musicmaker_one],
862         [(96, 8), (98, 8), sopranino_musicmaker_two],
863         [(98, 8), (100, 8), sopranino_musicmaker_two],
864         [(100, 8), (102, 8), sopranino_musicmaker_one],
865         [(102, 8), (104, 8), sopranino_musicmaker_one],
866         [(104, 8), (106, 8), sopranino_musicmaker_one],
867         [(106, 8), (108, 8), sopranino_musicmaker_one],
868         [(108, 8), (110, 8), sopranino_musicmaker_two],
869         [(110, 8), (112, 8), sopranino_musicmaker_two],
870         [(112, 8), (114, 8), sopranino_musicmaker_one],
```

```

871     [(114, 8), (116, 8), sopranino_musicmaker_one],
872     [(116, 8), (118, 8), sopranino_musicmaker_one],
873     [(118, 8), (120, 8), sopranino_musicmaker_one],
874     [(120, 8), (122, 8), sopranino_musicmaker_two],
875     [(122, 8), (124, 8), sopranino_musicmaker_two],
876     [(124, 8), (126, 8), sopranino_musicmaker_one],
877     [(126, 8), (128, 8), sopranino_musicmaker_one],
878     [(128, 8), (130, 8), sopranino_musicmaker_one],
879     [(130, 8), (132, 8), sopranino_musicmaker_one],
880     [(132, 8), (134, 8), sopranino_musicmaker_one],
881     [(134, 8), (136, 8), sopranino_musicmaker_two],
882     [(136, 8), (138, 8), sopranino_musicmaker_one],
883     [(138, 8), (140, 8), sopranino_musicmaker_one],
884     [(140, 8), (142, 8), sopranino_musicmaker_two],
885     [(142, 8), (144, 8), sopranino_musicmaker_two],
886     [(144, 8), (146, 8), sopranino_musicmaker_one],
887     [(146, 8), (148, 8), sopranino_musicmaker_one],
888     [(148, 8), (150, 8), sopranino_musicmaker_one],
889     [(150, 8), (152, 8), sopranino_musicmaker_two],
890     [(152, 8), (154, 8), sopranino_musicmaker_two],
891     [(154, 8), (156, 8), sopranino_musicmaker_one],
892     [(156, 8), (158, 8), sopranino_musicmaker_two],
893     [(158, 8), (160, 8), sopranino_musicmaker_two],
894     [(160, 8), (162, 8), sopranino_musicmaker_one],
895     [(162, 8), (164, 8), sopranino_musicmaker_two],
896     [(164, 8), (166, 8), sopranino_musicmaker_one],
897     [(166, 8), (168, 8), sopranino_musicmaker_one],
898     [(168, 8), (170, 8), sopranino_musicmaker_one],
899     [(170, 8), (172, 8), sopranino_musicmaker_one],
900     [(172, 8), (174, 8), sopranino_musicmaker_two],
901     [(174, 8), (176, 8), sopranino_musicmaker_one],
902     [(176, 8), (178, 8), sopranino_musicmaker_one],
903     [(178, 8), (180, 8), sopranino_musicmaker_one],
904     [(180, 8), (182, 8), sopranino_musicmaker_two],
905     [(182, 8), (184, 8), sopranino_musicmaker_one],
906     [(184, 8), (186, 8), sopranino_musicmaker_one],
907     [(186, 8), (188, 8), sopranino_musicmaker_one],
908     [(188, 8), (190, 8), sopranino_musicmaker_one],
909     [(190, 8), (192, 8), sopranino_musicmaker_one],
910     [(192, 8), (194, 8), sopranino_musicmaker_two],
911     [(194, 8), (196, 8), sopranino_musicmaker_one],
912     [(196, 8), (198, 8), sopranino_musicmaker_two],
913     [(198, 8), (199, 8), sopranino_musicmaker_one],
914     [(199, 8), (200, 8), sopranino_musicmaker_one],
915 ]
916 ])
917
918 voice_2_timespan_list = abjad.TimespanList([
919     abjad.AnnotatedTimespan(
920         start_offset=start_offset,
921         stop_offset=stop_offset,
922         annotation=MusicSpecifier(
923             music_maker=music_maker,
924             voice_name='Voice 2',
925         ),
926     )
927     for start_offset, stop_offset, music_maker in [
928         [(0, 8), (2, 8), soprano_one_musicmaker_one],
929         [(2, 8), (4, 8), soprano_one_musicmaker_one],

```

```
930      [(4, 8), (6, 8), soprano_one_musicmaker_two],  
931      [(6, 8), (8, 8), soprano_one_musicmaker_one],  
932      [(8, 8), (10, 8), soprano_one_musicmaker_one],  
933      [(10, 8), (12, 8), soprano_one_musicmaker_one],  
934      [(12, 8), (14, 8), soprano_one_musicmaker_two],  
935      [(14, 8), (16, 8), soprano_one_musicmaker_two],  
936      [(16, 8), (18, 8), soprano_one_musicmaker_one],  
937      [(18, 8), (20, 8), soprano_one_musicmaker_one],  
938      [(20, 8), (22, 8), soprano_one_musicmaker_one],  
939      [(22, 8), (24, 8), soprano_one_musicmaker_one],  
940      [(24, 8), (26, 8), soprano_one_musicmaker_one],  
941      [(26, 8), (28, 8), soprano_one_musicmaker_two],  
942      [(28, 8), (30, 8), soprano_one_musicmaker_two],  
943      [(30, 8), (32, 8), soprano_one_musicmaker_two],  
944      [(32, 8), (34, 8), soprano_one_musicmaker_one],  
945      [(34, 8), (36, 8), soprano_one_musicmaker_one],  
946      [(36, 8), (38, 8), soprano_one_musicmaker_one],  
947      [(38, 8), (40, 8), soprano_one_musicmaker_two],  
948      [(40, 8), (42, 8), soprano_one_musicmaker_two],  
949      [(42, 8), (44, 8), soprano_one_musicmaker_one],  
950      [(44, 8), (46, 8), soprano_one_musicmaker_one],  
951      [(46, 8), (48, 8), soprano_one_musicmaker_two],  
952      [(48, 8), (50, 8), soprano_one_musicmaker_one],  
953      [(50, 8), (52, 8), soprano_one_musicmaker_one],  
954      [(52, 8), (54, 8), soprano_one_musicmaker_one],  
955      [(54, 8), (56, 8), soprano_one_musicmaker_two],  
956      [(56, 8), (58, 8), soprano_one_musicmaker_one],  
957      [(58, 8), (60, 8), soprano_one_musicmaker_one],  
958      [(60, 8), (62, 8), soprano_one_musicmaker_one],  
959      [(62, 8), (64, 8), soprano_one_musicmaker_two],  
960      [(64, 8), (66, 8), soprano_one_musicmaker_two],  
961      [(66, 8), (68, 8), soprano_one_musicmaker_one],  
962      [(68, 8), (70, 8), soprano_one_musicmaker_two],  
963      [(70, 8), (72, 8), soprano_one_musicmaker_one],  
964      [(72, 8), (74, 8), soprano_one_musicmaker_one],  
965      [(74, 8), (76, 8), soprano_one_musicmaker_two],  
966      [(76, 8), (78, 8), soprano_one_musicmaker_two],  
967      [(78, 8), (80, 8), soprano_one_musicmaker_one],  
968      [(80, 8), (82, 8), soprano_one_musicmaker_two],  
969      [(82, 8), (84, 8), soprano_one_musicmaker_one],  
970      [(84, 8), (86, 8), soprano_one_musicmaker_two],  
971      [(86, 8), (88, 8), soprano_one_musicmaker_one],  
972      [(88, 8), (90, 8), soprano_one_musicmaker_one],  
973      [(90, 8), (92, 8), soprano_one_musicmaker_one],  
974      [(92, 8), (94, 8), soprano_one_musicmaker_two],  
975      [(94, 8), (96, 8), soprano_one_musicmaker_two],  
976      [(96, 8), (98, 8), soprano_one_musicmaker_one],  
977      [(98, 8), (100, 8), soprano_one_musicmaker_one],  
978      [(100, 8), (102, 8), soprano_one_musicmaker_one],  
979      [(102, 8), (104, 8), soprano_one_musicmaker_one],  
980      [(104, 8), (106, 8), soprano_one_musicmaker_two],  
981      [(106, 8), (108, 8), soprano_one_musicmaker_two],  
982      [(108, 8), (110, 8), soprano_one_musicmaker_one],  
983      [(110, 8), (112, 8), soprano_one_musicmaker_one],  
984      [(112, 8), (114, 8), soprano_one_musicmaker_one],  
985      [(114, 8), (116, 8), soprano_one_musicmaker_two],  
986      [(116, 8), (118, 8), soprano_one_musicmaker_two],  
987      [(118, 8), (120, 8), soprano_one_musicmaker_one],  
988      [(120, 8), (122, 8), soprano_one_musicmaker_one],
```

```

989     [(122, 8), (124, 8), soprano_one_musicmaker_two],
990     [(124, 8), (126, 8), soprano_one_musicmaker_two],
991     [(126, 8), (128, 8), soprano_one_musicmaker_one],
992     [(128, 8), (130, 8), soprano_one_musicmaker_one],
993     [(130, 8), (132, 8), soprano_one_musicmaker_one],
994     [(132, 8), (134, 8), soprano_one_musicmaker_two],
995     [(134, 8), (136, 8), soprano_one_musicmaker_two],
996     [(136, 8), (138, 8), soprano_one_musicmaker_one],
997     [(138, 8), (140, 8), soprano_one_musicmaker_one],
998     [(140, 8), (142, 8), soprano_one_musicmaker_two],
999     [(142, 8), (144, 8), soprano_one_musicmaker_one],
1000    [(144, 8), (146, 8), soprano_one_musicmaker_one],
1001    [(146, 8), (148, 8), soprano_one_musicmaker_two],
1002    [(148, 8), (150, 8), soprano_one_musicmaker_one],
1003    [(150, 8), (152, 8), soprano_one_musicmaker_one],
1004    [(152, 8), (154, 8), soprano_one_musicmaker_two],
1005    [(154, 8), (156, 8), soprano_one_musicmaker_one],
1006    [(156, 8), (158, 8), soprano_one_musicmaker_one],
1007    [(158, 8), (160, 8), soprano_one_musicmaker_one],
1008    [(160, 8), (162, 8), soprano_one_musicmaker_two],
1009    [(162, 8), (164, 8), soprano_one_musicmaker_one],
1010    [(164, 8), (166, 8), soprano_one_musicmaker_two],
1011    [(166, 8), (168, 8), soprano_one_musicmaker_one],
1012    [(168, 8), (170, 8), soprano_one_musicmaker_one],
1013    [(170, 8), (172, 8), soprano_one_musicmaker_two],
1014    [(172, 8), (174, 8), soprano_one_musicmaker_two],
1015    [(174, 8), (176, 8), soprano_one_musicmaker_one],
1016    [(176, 8), (178, 8), soprano_one_musicmaker_one],
1017    [(178, 8), (180, 8), soprano_one_musicmaker_one],
1018    [(180, 8), (182, 8), soprano_one_musicmaker_one],
1019    [(182, 8), (184, 8), soprano_one_musicmaker_two],
1020    [(184, 8), (186, 8), soprano_one_musicmaker_one],
1021    [(186, 8), (188, 8), soprano_one_musicmaker_one],
1022    [(188, 8), (190, 8), soprano_one_musicmaker_one],
1023    [(190, 8), (192, 8), soprano_one_musicmaker_two],
1024    [(192, 8), (194, 8), soprano_one_musicmaker_one],
1025    [(194, 8), (196, 8), soprano_one_musicmaker_one],
1026    [(196, 8), (198, 8), soprano_one_musicmaker_two],
1027    [(198, 8), (199, 8), soprano_one_musicmaker_one],
1028    [(199, 8), (200, 8), soprano_one_musicmaker_one],
1029  ]
1030 ])
1031
1032 voice_3_timespan_list = abjad.TimespanList([
1033     abjad.AnnotatedTimespan(
1034         start_offset=start_offset,
1035         stop_offset=stop_offset,
1036         annotation=MusicSpecifier(
1037             music_maker=music_maker,
1038             voice_name='Voice 3',
1039         ),
1040     )
1041     for start_offset, stop_offset, music_maker in [
1042         [(0, 8), (2, 8), soprano_two_musicmaker_one],
1043         [(2, 8), (4, 8), soprano_two_musicmaker_one],
1044         [(4, 8), (6, 8), soprano_two_musicmaker_one],
1045         [(6, 8), (8, 8), soprano_two_musicmaker_one],
1046         [(8, 8), (10, 8), soprano_two_musicmaker_one],
1047         [(10, 8), (12, 8), soprano_two_musicmaker_two],

```

```
1048      [(12, 8), (14, 8), soprano_two_musicmaker_one],  
1049      [(14, 8), (16, 8), soprano_two_musicmaker_two],  
1050      [(16, 8), (18, 8), soprano_two_musicmaker_two],  
1051      [(18, 8), (20, 8), soprano_two_musicmaker_one],  
1052      [(20, 8), (22, 8), soprano_two_musicmaker_one],  
1053      [(22, 8), (24, 8), soprano_two_musicmaker_two],  
1054      [(24, 8), (26, 8), soprano_two_musicmaker_two],  
1055      [(26, 8), (28, 8), soprano_two_musicmaker_one],  
1056      [(28, 8), (30, 8), soprano_two_musicmaker_one],  
1057      [(30, 8), (32, 8), soprano_two_musicmaker_two],  
1058      [(32, 8), (34, 8), soprano_two_musicmaker_one],  
1059      [(34, 8), (36, 8), soprano_two_musicmaker_one],  
1060      [(36, 8), (38, 8), soprano_two_musicmaker_two],  
1061      [(38, 8), (40, 8), soprano_two_musicmaker_two],  
1062      [(40, 8), (42, 8), soprano_two_musicmaker_two],  
1063      [(42, 8), (44, 8), soprano_two_musicmaker_one],  
1064      [(44, 8), (46, 8), soprano_two_musicmaker_one],  
1065      [(46, 8), (48, 8), soprano_two_musicmaker_one],  
1066      [(48, 8), (50, 8), soprano_two_musicmaker_one],  
1067      [(50, 8), (52, 8), soprano_two_musicmaker_two],  
1068      [(52, 8), (54, 8), soprano_two_musicmaker_one],  
1069      [(54, 8), (56, 8), soprano_two_musicmaker_one],  
1070      [(56, 8), (58, 8), soprano_two_musicmaker_one],  
1071      [(58, 8), (60, 8), soprano_two_musicmaker_two],  
1072      [(60, 8), (62, 8), soprano_two_musicmaker_two],  
1073      [(62, 8), (64, 8), soprano_two_musicmaker_one],  
1074      [(64, 8), (66, 8), soprano_two_musicmaker_one],  
1075      [(66, 8), (68, 8), soprano_two_musicmaker_one],  
1076      [(68, 8), (70, 8), soprano_two_musicmaker_two],  
1077      [(70, 8), (72, 8), soprano_two_musicmaker_one],  
1078      [(72, 8), (74, 8), soprano_two_musicmaker_one],  
1079      [(74, 8), (76, 8), soprano_two_musicmaker_two],  
1080      [(76, 8), (78, 8), soprano_two_musicmaker_one],  
1081      [(78, 8), (80, 8), soprano_two_musicmaker_two],  
1082      [(80, 8), (82, 8), soprano_two_musicmaker_one],  
1083      [(82, 8), (84, 8), soprano_two_musicmaker_one],  
1084      [(84, 8), (86, 8), soprano_two_musicmaker_two],  
1085      [(86, 8), (88, 8), soprano_two_musicmaker_two],  
1086      [(88, 8), (90, 8), soprano_two_musicmaker_two],  
1087      [(90, 8), (92, 8), soprano_two_musicmaker_one],  
1088      [(92, 8), (94, 8), soprano_two_musicmaker_two],  
1089      [(94, 8), (96, 8), soprano_two_musicmaker_two],  
1090      [(96, 8), (98, 8), soprano_two_musicmaker_one],  
1091      [(98, 8), (100, 8), soprano_two_musicmaker_one],  
1092      [(100, 8), (102, 8), soprano_two_musicmaker_two],  
1093      [(102, 8), (104, 8), soprano_two_musicmaker_one],  
1094      [(104, 8), (106, 8), soprano_two_musicmaker_two],  
1095      [(106, 8), (108, 8), soprano_two_musicmaker_one],  
1096      [(108, 8), (110, 8), soprano_two_musicmaker_two],  
1097      [(110, 8), (112, 8), soprano_two_musicmaker_one],  
1098      [(112, 8), (114, 8), soprano_two_musicmaker_two],  
1099      [(114, 8), (116, 8), soprano_two_musicmaker_one],  
1100      [(116, 8), (118, 8), soprano_two_musicmaker_two],  
1101      [(118, 8), (120, 8), soprano_two_musicmaker_one],  
1102      [(120, 8), (122, 8), soprano_two_musicmaker_one],  
1103      [(122, 8), (124, 8), soprano_two_musicmaker_two],  
1104      [(124, 8), (126, 8), soprano_two_musicmaker_two],  
1105      [(126, 8), (128, 8), soprano_two_musicmaker_one],  
1106      [(128, 8), (130, 8), soprano_two_musicmaker_one],
```

```

1107     [(130, 8), (132, 8), soprano_two_musicmaker_two],
1108     [(132, 8), (134, 8), soprano_two_musicmaker_one],
1109     [(134, 8), (136, 8), soprano_two_musicmaker_one],
1110     [(136, 8), (138, 8), soprano_two_musicmaker_one],
1111     [(138, 8), (140, 8), soprano_two_musicmaker_two],
1112     [(140, 8), (142, 8), soprano_two_musicmaker_one],
1113     [(142, 8), (144, 8), soprano_two_musicmaker_two],
1114     [(144, 8), (146, 8), soprano_two_musicmaker_one],
1115     [(146, 8), (148, 8), soprano_two_musicmaker_two],
1116     [(148, 8), (150, 8), soprano_two_musicmaker_two],
1117     [(150, 8), (152, 8), soprano_two_musicmaker_one],
1118     [(152, 8), (154, 8), soprano_two_musicmaker_two],
1119     [(154, 8), (156, 8), soprano_two_musicmaker_one],
1120     [(156, 8), (158, 8), soprano_two_musicmaker_one],
1121     [(158, 8), (160, 8), soprano_two_musicmaker_two],
1122     [(160, 8), (162, 8), soprano_two_musicmaker_one],
1123     [(162, 8), (164, 8), soprano_two_musicmaker_one],
1124     [(164, 8), (166, 8), soprano_two_musicmaker_one],
1125     [(166, 8), (168, 8), soprano_two_musicmaker_two],
1126     [(168, 8), (170, 8), soprano_two_musicmaker_two],
1127     [(170, 8), (172, 8), soprano_two_musicmaker_one],
1128     [(172, 8), (174, 8), soprano_two_musicmaker_one],
1129     [(174, 8), (176, 8), soprano_two_musicmaker_two],
1130     [(176, 8), (178, 8), soprano_two_musicmaker_two],
1131     [(178, 8), (180, 8), soprano_two_musicmaker_one],
1132     [(180, 8), (182, 8), soprano_two_musicmaker_one],
1133     [(182, 8), (184, 8), soprano_two_musicmaker_one],
1134     [(184, 8), (186, 8), soprano_two_musicmaker_two],
1135     [(186, 8), (188, 8), soprano_two_musicmaker_one],
1136     [(188, 8), (190, 8), soprano_two_musicmaker_one],
1137     [(190, 8), (192, 8), soprano_two_musicmaker_one],
1138     [(192, 8), (194, 8), soprano_two_musicmaker_two],
1139     [(194, 8), (196, 8), soprano_two_musicmaker_one],
1140     [(196, 8), (198, 8), soprano_two_musicmaker_one],
1141     [(198, 8), (199, 8), soprano_two_musicmaker_two],
1142     [(199, 8), (200, 8), soprano_two_musicmaker_one],
1143 ]
1144 ])
1145
1146 voice_4_timespan_list = abjad.TimespanList([
1147     abjad.AnnotatedTimespan(
1148         start_offset=start_offset,
1149         stop_offset=stop_offset,
1150         annotation=MusicSpecifier(
1151             music_maker=music_maker,
1152             voice_name='Voice 4',
1153         ),
1154     ),
1155     for start_offset, stop_offset, music_maker in [
1156         [(0, 8), (2, 8), soprano_three_musicmaker_one],
1157         [(2, 8), (4, 8), soprano_three_musicmaker_two],
1158         [(4, 8), (6, 8), soprano_three_musicmaker_two],
1159         [(6, 8), (8, 8), soprano_three_musicmaker_one],
1160         [(8, 8), (10, 8), soprano_three_musicmaker_two],
1161         [(10, 8), (12, 8), soprano_three_musicmaker_two],
1162         [(12, 8), (14, 8), soprano_three_musicmaker_one],
1163         [(14, 8), (16, 8), soprano_three_musicmaker_one],
1164         [(16, 8), (18, 8), soprano_three_musicmaker_two],
1165         [(18, 8), (20, 8), soprano_three_musicmaker_two],

```

```
1166 [(20, 8), (22, 8), soprano_three_musicmaker_one],  
1167 [(22, 8), (24, 8), soprano_three_musicmaker_one],  
1168 [(24, 8), (26, 8), soprano_three_musicmaker_two],  
1169 [(26, 8), (28, 8), soprano_three_musicmaker_two],  
1170 [(28, 8), (30, 8), soprano_three_musicmaker_one],  
1171 [(30, 8), (32, 8), soprano_three_musicmaker_two],  
1172 [(32, 8), (34, 8), soprano_three_musicmaker_one],  
1173 [(34, 8), (36, 8), soprano_three_musicmaker_one],  
1174 [(36, 8), (38, 8), soprano_three_musicmaker_one],  
1175 [(38, 8), (40, 8), soprano_three_musicmaker_two],  
1176 [(40, 8), (42, 8), soprano_three_musicmaker_one],  
1177 [(42, 8), (44, 8), soprano_three_musicmaker_one],  
1178 [(44, 8), (46, 8), soprano_three_musicmaker_one],  
1179 [(46, 8), (48, 8), soprano_three_musicmaker_two],  
1180 [(48, 8), (50, 8), soprano_three_musicmaker_one],  
1181 [(50, 8), (52, 8), soprano_three_musicmaker_one],  
1182 [(52, 8), (54, 8), soprano_three_musicmaker_one],  
1183 [(54, 8), (56, 8), soprano_three_musicmaker_two],  
1184 [(56, 8), (58, 8), soprano_three_musicmaker_one],  
1185 [(58, 8), (60, 8), soprano_three_musicmaker_one],  
1186 [(60, 8), (62, 8), soprano_three_musicmaker_one],  
1187 [(62, 8), (64, 8), soprano_three_musicmaker_two],  
1188 [(64, 8), (66, 8), soprano_three_musicmaker_two],  
1189 [(66, 8), (68, 8), soprano_three_musicmaker_two],  
1190 [(68, 8), (70, 8), soprano_three_musicmaker_one],  
1191 [(70, 8), (72, 8), soprano_three_musicmaker_one],  
1192 [(72, 8), (74, 8), soprano_three_musicmaker_one],  
1193 [(74, 8), (76, 8), soprano_three_musicmaker_two],  
1194 [(76, 8), (78, 8), soprano_three_musicmaker_one],  
1195 [(78, 8), (80, 8), soprano_three_musicmaker_two],  
1196 [(80, 8), (82, 8), soprano_three_musicmaker_one],  
1197 [(82, 8), (84, 8), soprano_three_musicmaker_one],  
1198 [(84, 8), (86, 8), soprano_three_musicmaker_one],  
1199 [(86, 8), (88, 8), soprano_three_musicmaker_one],  
1200 [(88, 8), (90, 8), soprano_three_musicmaker_one],  
1201 [(90, 8), (92, 8), soprano_three_musicmaker_two],  
1202 [(92, 8), (94, 8), soprano_three_musicmaker_two],  
1203 [(94, 8), (96, 8), soprano_three_musicmaker_two],  
1204 [(96, 8), (98, 8), soprano_three_musicmaker_one],  
1205 [(98, 8), (100, 8), soprano_three_musicmaker_one],  
1206 [(100, 8), (102, 8), soprano_three_musicmaker_two],  
1207 [(102, 8), (104, 8), soprano_three_musicmaker_two],  
1208 [(104, 8), (106, 8), soprano_three_musicmaker_one],  
1209 [(106, 8), (108, 8), soprano_three_musicmaker_two],  
1210 [(108, 8), (110, 8), soprano_three_musicmaker_two],  
1211 [(110, 8), (112, 8), soprano_three_musicmaker_one],  
1212 [(112, 8), (114, 8), soprano_three_musicmaker_two],  
1213 [(114, 8), (116, 8), soprano_three_musicmaker_one],  
1214 [(116, 8), (118, 8), soprano_three_musicmaker_two],  
1215 [(118, 8), (120, 8), soprano_three_musicmaker_one],  
1216 [(120, 8), (122, 8), soprano_three_musicmaker_one],  
1217 [(122, 8), (124, 8), soprano_three_musicmaker_one],  
1218 [(124, 8), (126, 8), soprano_three_musicmaker_two],  
1219 [(126, 8), (128, 8), soprano_three_musicmaker_one],  
1220 [(128, 8), (130, 8), soprano_three_musicmaker_one],  
1221 [(130, 8), (132, 8), soprano_three_musicmaker_two],  
1222 [(132, 8), (134, 8), soprano_three_musicmaker_two],  
1223 [(134, 8), (136, 8), soprano_three_musicmaker_one],  
1224 [(136, 8), (138, 8), soprano_three_musicmaker_one],
```

```

1225     [(138, 8), (140, 8), soprano_three_musicmaker_one],
1226     [(140, 8), (142, 8), soprano_three_musicmaker_two],
1227     [(142, 8), (144, 8), soprano_three_musicmaker_two],
1228     [(144, 8), (146, 8), soprano_three_musicmaker_two],
1229     [(146, 8), (148, 8), soprano_three_musicmaker_one],
1230     [(148, 8), (150, 8), soprano_three_musicmaker_one],
1231     [(150, 8), (152, 8), soprano_three_musicmaker_one],
1232     [(152, 8), (154, 8), soprano_three_musicmaker_one],
1233     [(154, 8), (156, 8), soprano_three_musicmaker_one],
1234     [(156, 8), (158, 8), soprano_three_musicmaker_two],
1235     [(158, 8), (160, 8), soprano_three_musicmaker_two],
1236     [(160, 8), (162, 8), soprano_three_musicmaker_one],
1237     [(162, 8), (164, 8), soprano_three_musicmaker_one],
1238     [(164, 8), (166, 8), soprano_three_musicmaker_one],
1239     [(166, 8), (168, 8), soprano_three_musicmaker_one],
1240     [(168, 8), (170, 8), soprano_three_musicmaker_two],
1241     [(170, 8), (172, 8), soprano_three_musicmaker_two],
1242     [(172, 8), (174, 8), soprano_three_musicmaker_one],
1243     [(174, 8), (176, 8), soprano_three_musicmaker_one],
1244     [(176, 8), (178, 8), soprano_three_musicmaker_one],
1245     [(178, 8), (180, 8), soprano_three_musicmaker_two],
1246     [(180, 8), (182, 8), soprano_three_musicmaker_one],
1247     [(182, 8), (184, 8), soprano_three_musicmaker_one],
1248     [(184, 8), (186, 8), soprano_three_musicmaker_one],
1249     [(186, 8), (188, 8), soprano_three_musicmaker_one],
1250     [(188, 8), (190, 8), soprano_three_musicmaker_two],
1251     [(190, 8), (192, 8), soprano_three_musicmaker_two],
1252     [(192, 8), (194, 8), soprano_three_musicmaker_one],
1253     [(194, 8), (196, 8), soprano_three_musicmaker_one],
1254     [(196, 8), (198, 8), soprano_three_musicmaker_two],
1255     [(198, 8), (199, 8), soprano_three_musicmaker_one],
1256     [(199, 8), (200, 8), soprano_three_musicmaker_one],
1257 ]
1258 ])
1259
1260 voice_5_timespan_list = abjad.TimespanList([
1261     abjad.AnnotatedTimespan(
1262         start_offset=start_offset,
1263         stop_offset=stop_offset,
1264         annotation=MusicSpecifier(
1265             music_maker=music_maker,
1266             voice_name='Voice 5',
1267         ),
1268     )
1269     for start_offset, stop_offset, music_maker in [
1270         [(0, 8), (2, 8), alto_one_musicmaker_two],
1271         [(2, 8), (4, 8), alto_one_musicmaker_one],
1272         [(4, 8), (6, 8), alto_one_musicmaker_two],
1273         [(6, 8), (8, 8), alto_one_musicmaker_one],
1274         [(8, 8), (10, 8), alto_one_musicmaker_two],
1275         [(10, 8), (12, 8), alto_one_musicmaker_one],
1276         [(12, 8), (14, 8), alto_one_musicmaker_one],
1277         [(14, 8), (16, 8), alto_one_musicmaker_two],
1278         [(16, 8), (18, 8), alto_one_musicmaker_one],
1279         [(18, 8), (20, 8), alto_one_musicmaker_two],
1280         [(20, 8), (22, 8), alto_one_musicmaker_one],
1281         [(22, 8), (24, 8), alto_one_musicmaker_two],
1282         [(24, 8), (26, 8), alto_one_musicmaker_two],
1283         [(26, 8), (28, 8), alto_one_musicmaker_one],

```

```
1284      [(28, 8), (30, 8), alto_one_musicmaker_one],  
1285      [(30, 8), (32, 8), alto_one_musicmaker_one],  
1286      [(32, 8), (34, 8), alto_one_musicmaker_two],  
1287      [(34, 8), (36, 8), alto_one_musicmaker_two],  
1288      [(36, 8), (38, 8), alto_one_musicmaker_one],  
1289      [(38, 8), (40, 8), alto_one_musicmaker_one],  
1290      [(40, 8), (42, 8), alto_one_musicmaker_one],  
1291      [(42, 8), (44, 8), alto_one_musicmaker_two],  
1292      [(44, 8), (46, 8), alto_one_musicmaker_two],  
1293      [(46, 8), (48, 8), alto_one_musicmaker_one],  
1294      [(48, 8), (50, 8), alto_one_musicmaker_one],  
1295      [(50, 8), (52, 8), alto_one_musicmaker_one],  
1296      [(52, 8), (54, 8), alto_one_musicmaker_two],  
1297      [(54, 8), (56, 8), alto_one_musicmaker_two],  
1298      [(56, 8), (58, 8), alto_one_musicmaker_one],  
1299      [(58, 8), (60, 8), alto_one_musicmaker_two],  
1300      [(60, 8), (62, 8), alto_one_musicmaker_two],  
1301      [(62, 8), (64, 8), alto_one_musicmaker_one],  
1302      [(64, 8), (66, 8), alto_one_musicmaker_one],  
1303      [(66, 8), (68, 8), alto_one_musicmaker_one],  
1304      [(68, 8), (70, 8), alto_one_musicmaker_two],  
1305      [(70, 8), (72, 8), alto_one_musicmaker_two],  
1306      [(72, 8), (74, 8), alto_one_musicmaker_one],  
1307      [(74, 8), (76, 8), alto_one_musicmaker_two],  
1308      [(76, 8), (78, 8), alto_one_musicmaker_one],  
1309      [(78, 8), (80, 8), alto_one_musicmaker_two],  
1310      [(80, 8), (82, 8), alto_one_musicmaker_two],  
1311      [(82, 8), (84, 8), alto_one_musicmaker_one],  
1312      [(84, 8), (86, 8), alto_one_musicmaker_one],  
1313      [(86, 8), (88, 8), alto_one_musicmaker_one],  
1314      [(88, 8), (90, 8), alto_one_musicmaker_one],  
1315      [(90, 8), (92, 8), alto_one_musicmaker_one],  
1316      [(92, 8), (94, 8), alto_one_musicmaker_one],  
1317      [(94, 8), (96, 8), alto_one_musicmaker_two],  
1318      [(96, 8), (98, 8), alto_one_musicmaker_one],  
1319      [(98, 8), (100, 8), alto_one_musicmaker_one],  
1320      [(100, 8), (102, 8), alto_one_musicmaker_one],  
1321      [(102, 8), (104, 8), alto_one_musicmaker_one],  
1322      [(104, 8), (106, 8), alto_one_musicmaker_two],  
1323      [(106, 8), (108, 8), alto_one_musicmaker_two],  
1324      [(108, 8), (110, 8), alto_one_musicmaker_one],  
1325      [(110, 8), (112, 8), alto_one_musicmaker_one],  
1326      [(112, 8), (114, 8), alto_one_musicmaker_one],  
1327      [(114, 8), (116, 8), alto_one_musicmaker_one],  
1328      [(116, 8), (118, 8), alto_one_musicmaker_two],  
1329      [(118, 8), (120, 8), alto_one_musicmaker_one],  
1330      [(120, 8), (122, 8), alto_one_musicmaker_one],  
1331      [(122, 8), (124, 8), alto_one_musicmaker_one],  
1332      [(124, 8), (126, 8), alto_one_musicmaker_one],  
1333      [(126, 8), (128, 8), alto_one_musicmaker_one],  
1334      [(128, 8), (130, 8), alto_one_musicmaker_two],  
1335      [(130, 8), (132, 8), alto_one_musicmaker_two],  
1336      [(132, 8), (134, 8), alto_one_musicmaker_one],  
1337      [(134, 8), (136, 8), alto_one_musicmaker_one],  
1338      [(136, 8), (138, 8), alto_one_musicmaker_two],  
1339      [(138, 8), (140, 8), alto_one_musicmaker_two],  
1340      [(140, 8), (142, 8), alto_one_musicmaker_one],  
1341      [(142, 8), (144, 8), alto_one_musicmaker_one],  
1342      [(144, 8), (146, 8), alto_one_musicmaker_one],
```

```

1343     [(146, 8), (148, 8), alto_one_musicmaker_two],
1344     [(148, 8), (150, 8), alto_one_musicmaker_one],
1345     [(150, 8), (152, 8), alto_one_musicmaker_one],
1346     [(152, 8), (154, 8), alto_one_musicmaker_one],
1347     [(154, 8), (156, 8), alto_one_musicmaker_one],
1348     [(156, 8), (158, 8), alto_one_musicmaker_two],
1349     [(158, 8), (160, 8), alto_one_musicmaker_two],
1350     [(160, 8), (162, 8), alto_one_musicmaker_one],
1351     [(162, 8), (164, 8), alto_one_musicmaker_one],
1352     [(164, 8), (166, 8), alto_one_musicmaker_one],
1353     [(166, 8), (168, 8), alto_one_musicmaker_two],
1354     [(168, 8), (170, 8), alto_one_musicmaker_one],
1355     [(170, 8), (172, 8), alto_one_musicmaker_two],
1356     [(172, 8), (174, 8), alto_one_musicmaker_one],
1357     [(174, 8), (176, 8), alto_one_musicmaker_one],
1358     [(176, 8), (178, 8), alto_one_musicmaker_one],
1359     [(178, 8), (180, 8), alto_one_musicmaker_two],
1360     [(180, 8), (182, 8), alto_one_musicmaker_one],
1361     [(182, 8), (184, 8), alto_one_musicmaker_one],
1362     [(184, 8), (186, 8), alto_one_musicmaker_one],
1363     [(186, 8), (188, 8), alto_one_musicmaker_two],
1364     [(188, 8), (190, 8), alto_one_musicmaker_one],
1365     [(190, 8), (192, 8), alto_one_musicmaker_one],
1366     [(192, 8), (194, 8), alto_one_musicmaker_two],
1367     [(194, 8), (196, 8), alto_one_musicmaker_two],
1368     [(196, 8), (198, 8), alto_one_musicmaker_one],
1369     [(198, 8), (199, 8), alto_one_musicmaker_one],
1370     [(199, 8), (200, 8), alto_one_musicmaker_one],
1371 ]
1372 ])
1373
1374 voice_6_timespan_list = abjad.TimespanList([
1375     abjad.AnnotatedTimespan(
1376         start_offset=start_offset,
1377         stop_offset=stop_offset,
1378         annotation=MusicSpecifier(
1379             music_maker=music_maker,
1380             voice_name='Voice 6',
1381         ),
1382     )
1383     for start_offset, stop_offset, music_maker in [
1384         [(0, 8), (2, 8), alto_two_musicmaker_one],
1385         [(2, 8), (4, 8), alto_two_musicmaker_one],
1386         [(4, 8), (6, 8), alto_two_musicmaker_one],
1387         [(6, 8), (8, 8), alto_two_musicmaker_one],
1388         [(8, 8), (10, 8), alto_two_musicmaker_one],
1389         [(10, 8), (12, 8), alto_two_musicmaker_two],
1390         [(12, 8), (14, 8), alto_two_musicmaker_two],
1391         [(14, 8), (16, 8), alto_two_musicmaker_one],
1392         [(16, 8), (18, 8), alto_two_musicmaker_one],
1393         [(18, 8), (20, 8), alto_two_musicmaker_two],
1394         [(20, 8), (22, 8), alto_two_musicmaker_two],
1395         [(22, 8), (24, 8), alto_two_musicmaker_two],
1396         [(24, 8), (26, 8), alto_two_musicmaker_one],
1397         [(26, 8), (28, 8), alto_two_musicmaker_one],
1398         [(28, 8), (30, 8), alto_two_musicmaker_one],
1399         [(30, 8), (32, 8), alto_two_musicmaker_two],
1400         [(32, 8), (34, 8), alto_two_musicmaker_one],
1401         [(34, 8), (36, 8), alto_two_musicmaker_one],

```

```
1402      [(36, 8), (38, 8), alto_two_musicmaker_one],  
1403      [(38, 8), (40, 8), alto_two_musicmaker_two],  
1404      [(40, 8), (42, 8), alto_two_musicmaker_two],  
1405      [(42, 8), (44, 8), alto_two_musicmaker_two],  
1406      [(44, 8), (46, 8), alto_two_musicmaker_one],  
1407      [(46, 8), (48, 8), alto_two_musicmaker_one],  
1408      [(48, 8), (50, 8), alto_two_musicmaker_one],  
1409      [(50, 8), (52, 8), alto_two_musicmaker_one],  
1410      [(52, 8), (54, 8), alto_two_musicmaker_one],  
1411      [(54, 8), (56, 8), alto_two_musicmaker_two],  
1412      [(56, 8), (58, 8), alto_two_musicmaker_two],  
1413      [(58, 8), (60, 8), alto_two_musicmaker_one],  
1414      [(60, 8), (62, 8), alto_two_musicmaker_one],  
1415      [(62, 8), (64, 8), alto_two_musicmaker_one],  
1416      [(64, 8), (66, 8), alto_two_musicmaker_one],  
1417      [(66, 8), (68, 8), alto_two_musicmaker_two],  
1418      [(68, 8), (70, 8), alto_two_musicmaker_one],  
1419      [(70, 8), (72, 8), alto_two_musicmaker_one],  
1420      [(72, 8), (74, 8), alto_two_musicmaker_two],  
1421      [(74, 8), (76, 8), alto_two_musicmaker_two],  
1422      [(76, 8), (78, 8), alto_two_musicmaker_one],  
1423      [(78, 8), (80, 8), alto_two_musicmaker_one],  
1424      [(80, 8), (82, 8), alto_two_musicmaker_one],  
1425      [(82, 8), (84, 8), alto_two_musicmaker_two],  
1426      [(84, 8), (86, 8), alto_two_musicmaker_two],  
1427      [(86, 8), (88, 8), alto_two_musicmaker_one],  
1428      [(88, 8), (90, 8), alto_two_musicmaker_one],  
1429      [(90, 8), (92, 8), alto_two_musicmaker_one],  
1430      [(92, 8), (94, 8), alto_two_musicmaker_two],  
1431      [(94, 8), (96, 8), alto_two_musicmaker_one],  
1432      [(96, 8), (98, 8), alto_two_musicmaker_two],  
1433      [(98, 8), (100, 8), alto_two_musicmaker_two],  
1434      [(100, 8), (102, 8), alto_two_musicmaker_one],  
1435      [(102, 8), (104, 8), alto_two_musicmaker_one],  
1436      [(104, 8), (106, 8), alto_two_musicmaker_two],  
1437      [(106, 8), (108, 8), alto_two_musicmaker_one],  
1438      [(108, 8), (110, 8), alto_two_musicmaker_one],  
1439      [(110, 8), (112, 8), alto_two_musicmaker_two],  
1440      [(112, 8), (114, 8), alto_two_musicmaker_one],  
1441      [(114, 8), (116, 8), alto_two_musicmaker_one],  
1442      [(116, 8), (118, 8), alto_two_musicmaker_two],  
1443      [(118, 8), (120, 8), alto_two_musicmaker_two],  
1444      [(120, 8), (122, 8), alto_two_musicmaker_one],  
1445      [(122, 8), (124, 8), alto_two_musicmaker_one],  
1446      [(124, 8), (126, 8), alto_two_musicmaker_one],  
1447      [(126, 8), (128, 8), alto_two_musicmaker_two],  
1448      [(128, 8), (130, 8), alto_two_musicmaker_two],  
1449      [(130, 8), (132, 8), alto_two_musicmaker_two],  
1450      [(132, 8), (134, 8), alto_two_musicmaker_one],  
1451      [(134, 8), (136, 8), alto_two_musicmaker_one],  
1452      [(136, 8), (138, 8), alto_two_musicmaker_one],  
1453      [(138, 8), (140, 8), alto_two_musicmaker_one],  
1454      [(140, 8), (142, 8), alto_two_musicmaker_one],  
1455      [(142, 8), (144, 8), alto_two_musicmaker_two],  
1456      [(144, 8), (146, 8), alto_two_musicmaker_one],  
1457      [(146, 8), (148, 8), alto_two_musicmaker_two],  
1458      [(148, 8), (150, 8), alto_two_musicmaker_two],  
1459      [(150, 8), (152, 8), alto_two_musicmaker_one],  
1460      [(152, 8), (154, 8), alto_two_musicmaker_one],
```

```

1461     [(154, 8), (156, 8), alto_two_musicmaker_one],
1462     [(156, 8), (158, 8), alto_two_musicmaker_two],
1463     [(158, 8), (160, 8), alto_two_musicmaker_one],
1464     [(160, 8), (162, 8), alto_two_musicmaker_one],
1465     [(162, 8), (164, 8), alto_two_musicmaker_one],
1466     [(164, 8), (166, 8), alto_two_musicmaker_two],
1467     [(166, 8), (168, 8), alto_two_musicmaker_two],
1468     [(168, 8), (170, 8), alto_two_musicmaker_one],
1469     [(170, 8), (172, 8), alto_two_musicmaker_one],
1470     [(172, 8), (174, 8), alto_two_musicmaker_two],
1471     [(174, 8), (176, 8), alto_two_musicmaker_one],
1472     [(176, 8), (178, 8), alto_two_musicmaker_one],
1473     [(178, 8), (180, 8), alto_two_musicmaker_one],
1474     [(180, 8), (182, 8), alto_two_musicmaker_two],
1475     [(182, 8), (184, 8), alto_two_musicmaker_two],
1476     [(184, 8), (186, 8), alto_two_musicmaker_one],
1477     [(186, 8), (188, 8), alto_two_musicmaker_one],
1478     [(188, 8), (190, 8), alto_two_musicmaker_two],
1479     [(190, 8), (192, 8), alto_two_musicmaker_one],
1480     [(192, 8), (194, 8), alto_two_musicmaker_one],
1481     [(194, 8), (196, 8), alto_two_musicmaker_two],
1482     [(196, 8), (198, 8), alto_two_musicmaker_two],
1483     [(198, 8), (199, 8), alto_two_musicmaker_one],
1484     [(199, 8), (200, 8), alto_two_musicmaker_one],
1485   ]
1486 ])
1487
1488 voice_7_timespan_list = abjad.TimespanList([
1489     abjad.AnnotatedTimespan(
1490         start_offset=start_offset,
1491         stop_offset=stop_offset,
1492         annotation=MusicSpecifier(
1493             music_maker=music_maker,
1494             voice_name='Voice 7',
1495         ),
1496     ),
1497     for start_offset, stop_offset, music_maker in [
1498         [(0, 8), (2, 8), alto_three_musicmaker_one],
1499         [(2, 8), (4, 8), alto_three_musicmaker_one],
1500         [(4, 8), (6, 8), alto_three_musicmaker_two],
1501         [(6, 8), (8, 8), alto_three_musicmaker_two],
1502         [(8, 8), (10, 8), alto_three_musicmaker_one],
1503         [(10, 8), (12, 8), alto_three_musicmaker_two],
1504         [(12, 8), (14, 8), alto_three_musicmaker_two],
1505         [(14, 8), (16, 8), alto_three_musicmaker_one],
1506         [(16, 8), (18, 8), alto_three_musicmaker_one],
1507         [(18, 8), (20, 8), alto_three_musicmaker_one],
1508         [(20, 8), (22, 8), alto_three_musicmaker_one],
1509         [(22, 8), (24, 8), alto_three_musicmaker_one],
1510         [(24, 8), (26, 8), alto_three_musicmaker_two],
1511         [(26, 8), (28, 8), alto_three_musicmaker_two],
1512         [(28, 8), (30, 8), alto_three_musicmaker_one],
1513         [(30, 8), (32, 8), alto_three_musicmaker_one],
1514         [(32, 8), (34, 8), alto_three_musicmaker_one],
1515         [(34, 8), (36, 8), alto_three_musicmaker_two],
1516         [(36, 8), (38, 8), alto_three_musicmaker_two],
1517         [(38, 8), (40, 8), alto_three_musicmaker_one],
1518         [(40, 8), (42, 8), alto_three_musicmaker_one],
1519         [(42, 8), (44, 8), alto_three_musicmaker_one],

```

```
1520      [(44, 8), (46, 8), alto_three_musicmaker_one],  
1521      [(46, 8), (48, 8), alto_three_musicmaker_two],  
1522      [(48, 8), (50, 8), alto_three_musicmaker_one],  
1523      [(50, 8), (52, 8), alto_three_musicmaker_one],  
1524      [(52, 8), (54, 8), alto_three_musicmaker_one],  
1525      [(54, 8), (56, 8), alto_three_musicmaker_two],  
1526      [(56, 8), (58, 8), alto_three_musicmaker_one],  
1527      [(58, 8), (60, 8), alto_three_musicmaker_one],  
1528      [(60, 8), (62, 8), alto_three_musicmaker_one],  
1529      [(62, 8), (64, 8), alto_three_musicmaker_one],  
1530      [(64, 8), (66, 8), alto_three_musicmaker_two],  
1531      [(66, 8), (68, 8), alto_three_musicmaker_one],  
1532      [(68, 8), (70, 8), alto_three_musicmaker_two],  
1533      [(70, 8), (72, 8), alto_three_musicmaker_one],  
1534      [(72, 8), (74, 8), alto_three_musicmaker_two],  
1535      [(74, 8), (76, 8), alto_three_musicmaker_two],  
1536      [(76, 8), (78, 8), alto_three_musicmaker_one],  
1537      [(78, 8), (80, 8), alto_three_musicmaker_one],  
1538      [(80, 8), (82, 8), alto_three_musicmaker_one],  
1539      [(82, 8), (84, 8), alto_three_musicmaker_two],  
1540      [(84, 8), (86, 8), alto_three_musicmaker_one],  
1541      [(86, 8), (88, 8), alto_three_musicmaker_two],  
1542      [(88, 8), (90, 8), alto_three_musicmaker_one],  
1543      [(90, 8), (92, 8), alto_three_musicmaker_two],  
1544      [(92, 8), (94, 8), alto_three_musicmaker_one],  
1545      [(94, 8), (96, 8), alto_three_musicmaker_two],  
1546      [(96, 8), (98, 8), alto_three_musicmaker_one],  
1547      [(98, 8), (100, 8), alto_three_musicmaker_two],  
1548      [(100, 8), (102, 8), alto_three_musicmaker_two],  
1549      [(102, 8), (104, 8), alto_three_musicmaker_one],  
1550      [(104, 8), (106, 8), alto_three_musicmaker_one],  
1551      [(106, 8), (108, 8), alto_three_musicmaker_two],  
1552      [(108, 8), (110, 8), alto_three_musicmaker_one],  
1553      [(110, 8), (112, 8), alto_three_musicmaker_one],  
1554      [(112, 8), (114, 8), alto_three_musicmaker_one],  
1555      [(114, 8), (116, 8), alto_three_musicmaker_two],  
1556      [(116, 8), (118, 8), alto_three_musicmaker_one],  
1557      [(118, 8), (120, 8), alto_three_musicmaker_one],  
1558      [(120, 8), (122, 8), alto_three_musicmaker_two],  
1559      [(122, 8), (124, 8), alto_three_musicmaker_two],  
1560      [(124, 8), (126, 8), alto_three_musicmaker_one],  
1561      [(126, 8), (128, 8), alto_three_musicmaker_one],  
1562      [(128, 8), (130, 8), alto_three_musicmaker_two],  
1563      [(130, 8), (132, 8), alto_three_musicmaker_one],  
1564      [(132, 8), (134, 8), alto_three_musicmaker_two],  
1565      [(134, 8), (136, 8), alto_three_musicmaker_one],  
1566      [(136, 8), (138, 8), alto_three_musicmaker_two],  
1567      [(138, 8), (140, 8), alto_three_musicmaker_one],  
1568      [(140, 8), (142, 8), alto_three_musicmaker_one],  
1569      [(142, 8), (144, 8), alto_three_musicmaker_one],  
1570      [(144, 8), (146, 8), alto_three_musicmaker_one],  
1571      [(146, 8), (148, 8), alto_three_musicmaker_two],  
1572      [(148, 8), (150, 8), alto_three_musicmaker_two],  
1573      [(150, 8), (152, 8), alto_three_musicmaker_two],  
1574      [(152, 8), (154, 8), alto_three_musicmaker_one],  
1575      [(154, 8), (156, 8), alto_three_musicmaker_one],  
1576      [(156, 8), (158, 8), alto_three_musicmaker_one],  
1577      [(158, 8), (160, 8), alto_three_musicmaker_one],  
1578      [(160, 8), (162, 8), alto_three_musicmaker_two],
```

```

1579     [(162, 8), (164, 8), alto_three_musicmaker_two],
1580     [(164, 8), (166, 8), alto_three_musicmaker_one],
1581     [(166, 8), (168, 8), alto_three_musicmaker_one],
1582     [(168, 8), (170, 8), alto_three_musicmaker_two],
1583     [(170, 8), (172, 8), alto_three_musicmaker_two],
1584     [(172, 8), (174, 8), alto_three_musicmaker_one],
1585     [(174, 8), (176, 8), alto_three_musicmaker_one],
1586     [(176, 8), (178, 8), alto_three_musicmaker_one],
1587     [(178, 8), (180, 8), alto_three_musicmaker_two],
1588     [(180, 8), (182, 8), alto_three_musicmaker_one],
1589     [(182, 8), (184, 8), alto_three_musicmaker_two],
1590     [(184, 8), (186, 8), alto_three_musicmaker_one],
1591     [(186, 8), (188, 8), alto_three_musicmaker_one],
1592     [(188, 8), (190, 8), alto_three_musicmaker_one],
1593     [(190, 8), (192, 8), alto_three_musicmaker_one],
1594     [(192, 8), (194, 8), alto_three_musicmaker_two],
1595     [(194, 8), (196, 8), alto_three_musicmaker_two],
1596     [(196, 8), (198, 8), alto_three_musicmaker_one],
1597     [(198, 8), (199, 8), alto_three_musicmaker_one],
1598     [(199, 8), (200, 8), alto_three_musicmaker_two],
1599   ]
1600 ])
1601
1602 voice_8_timespan_list = abjad.TimespanList([
1603     abjad.AnnotatedTimespan(
1604         start_offset=start_offset,
1605         stop_offset=stop_offset,
1606         annotation=MusicSpecifier(
1607             music_maker=music_maker,
1608             voice_name='Voice 8',
1609         ),
1610     ),
1611     for start_offset, stop_offset, music_maker in [
1612         [(0, 8), (2, 8), alto_four_musicmaker_one],
1613         [(2, 8), (4, 8), alto_four_musicmaker_one],
1614         [(4, 8), (6, 8), alto_four_musicmaker_one],
1615         [(6, 8), (8, 8), alto_four_musicmaker_two],
1616         [(8, 8), (10, 8), alto_four_musicmaker_one],
1617         [(10, 8), (12, 8), alto_four_musicmaker_one],
1618         [(12, 8), (14, 8), alto_four_musicmaker_two],
1619         [(14, 8), (16, 8), alto_four_musicmaker_one],
1620         [(16, 8), (18, 8), alto_four_musicmaker_one],
1621         [(18, 8), (20, 8), alto_four_musicmaker_one],
1622         [(20, 8), (22, 8), alto_four_musicmaker_two],
1623         [(22, 8), (24, 8), alto_four_musicmaker_one],
1624         [(24, 8), (26, 8), alto_four_musicmaker_two],
1625         [(26, 8), (28, 8), alto_four_musicmaker_two],
1626         [(28, 8), (30, 8), alto_four_musicmaker_one],
1627         [(30, 8), (32, 8), alto_four_musicmaker_one],
1628         [(32, 8), (34, 8), alto_four_musicmaker_one],
1629         [(34, 8), (36, 8), alto_four_musicmaker_one],
1630         [(36, 8), (38, 8), alto_four_musicmaker_two],
1631         [(38, 8), (40, 8), alto_four_musicmaker_two],
1632         [(40, 8), (42, 8), alto_four_musicmaker_two],
1633         [(42, 8), (44, 8), alto_four_musicmaker_one],
1634         [(44, 8), (46, 8), alto_four_musicmaker_one],
1635         [(46, 8), (48, 8), alto_four_musicmaker_one],
1636         [(48, 8), (50, 8), alto_four_musicmaker_one],
1637         [(50, 8), (52, 8), alto_four_musicmaker_one],

```

```
1638      [(52, 8), (54, 8), alto_four_musicmaker_one],  
1639      [(54, 8), (56, 8), alto_four_musicmaker_one],  
1640      [(56, 8), (58, 8), alto_four_musicmaker_two],  
1641      [(58, 8), (60, 8), alto_four_musicmaker_two],  
1642      [(60, 8), (62, 8), alto_four_musicmaker_one],  
1643      [(62, 8), (64, 8), alto_four_musicmaker_two],  
1644      [(64, 8), (66, 8), alto_four_musicmaker_one],  
1645      [(66, 8), (68, 8), alto_four_musicmaker_one],  
1646      [(68, 8), (70, 8), alto_four_musicmaker_one],  
1647      [(70, 8), (72, 8), alto_four_musicmaker_one],  
1648      [(72, 8), (74, 8), alto_four_musicmaker_two],  
1649      [(74, 8), (76, 8), alto_four_musicmaker_two],  
1650      [(76, 8), (78, 8), alto_four_musicmaker_one],  
1651      [(78, 8), (80, 8), alto_four_musicmaker_one],  
1652      [(80, 8), (82, 8), alto_four_musicmaker_one],  
1653      [(82, 8), (84, 8), alto_four_musicmaker_one],  
1654      [(84, 8), (86, 8), alto_four_musicmaker_one],  
1655      [(86, 8), (88, 8), alto_four_musicmaker_two],  
1656      [(88, 8), (90, 8), alto_four_musicmaker_one],  
1657      [(90, 8), (92, 8), alto_four_musicmaker_one],  
1658      [(92, 8), (94, 8), alto_four_musicmaker_two],  
1659      [(94, 8), (96, 8), alto_four_musicmaker_one],  
1660      [(96, 8), (98, 8), alto_four_musicmaker_one],  
1661      [(98, 8), (100, 8), alto_four_musicmaker_one],  
1662      [(100, 8), (102, 8), alto_four_musicmaker_two],  
1663      [(102, 8), (104, 8), alto_four_musicmaker_two],  
1664      [(104, 8), (106, 8), alto_four_musicmaker_one],  
1665      [(106, 8), (108, 8), alto_four_musicmaker_one],  
1666      [(108, 8), (110, 8), alto_four_musicmaker_one],  
1667      [(110, 8), (112, 8), alto_four_musicmaker_one],  
1668      [(112, 8), (114, 8), alto_four_musicmaker_two],  
1669      [(114, 8), (116, 8), alto_four_musicmaker_two],  
1670      [(116, 8), (118, 8), alto_four_musicmaker_one],  
1671      [(118, 8), (120, 8), alto_four_musicmaker_one],  
1672      [(120, 8), (122, 8), alto_four_musicmaker_one],  
1673      [(122, 8), (124, 8), alto_four_musicmaker_two],  
1674      [(124, 8), (126, 8), alto_four_musicmaker_two],  
1675      [(126, 8), (128, 8), alto_four_musicmaker_one],  
1676      [(128, 8), (130, 8), alto_four_musicmaker_one],  
1677      [(130, 8), (132, 8), alto_four_musicmaker_one],  
1678      [(132, 8), (134, 8), alto_four_musicmaker_two],  
1679      [(134, 8), (136, 8), alto_four_musicmaker_two],  
1680      [(136, 8), (138, 8), alto_four_musicmaker_two],  
1681      [(138, 8), (140, 8), alto_four_musicmaker_one],  
1682      [(140, 8), (142, 8), alto_four_musicmaker_one],  
1683      [(142, 8), (144, 8), alto_four_musicmaker_one],  
1684      [(144, 8), (146, 8), alto_four_musicmaker_one],  
1685      [(146, 8), (148, 8), alto_four_musicmaker_one],  
1686      [(148, 8), (150, 8), alto_four_musicmaker_one],  
1687      [(150, 8), (152, 8), alto_four_musicmaker_one],  
1688      [(152, 8), (154, 8), alto_four_musicmaker_two],  
1689      [(154, 8), (156, 8), alto_four_musicmaker_two],  
1690      [(156, 8), (158, 8), alto_four_musicmaker_one],  
1691      [(158, 8), (160, 8), alto_four_musicmaker_one],  
1692      [(160, 8), (162, 8), alto_four_musicmaker_one],  
1693      [(162, 8), (164, 8), alto_four_musicmaker_two],  
1694      [(164, 8), (166, 8), alto_four_musicmaker_two],  
1695      [(166, 8), (168, 8), alto_four_musicmaker_one],  
1696      [(168, 8), (170, 8), alto_four_musicmaker_one],
```

```

1697     [(170, 8), (172, 8), alto_four_musicmaker_one],
1698     [(172, 8), (174, 8), alto_four_musicmaker_one],
1699     [(174, 8), (176, 8), alto_four_musicmaker_one],
1700     [(176, 8), (178, 8), alto_four_musicmaker_one],
1701     [(178, 8), (180, 8), alto_four_musicmaker_two],
1702     [(180, 8), (182, 8), alto_four_musicmaker_two],
1703     [(182, 8), (184, 8), alto_four_musicmaker_one],
1704     [(184, 8), (186, 8), alto_four_musicmaker_two],
1705     [(186, 8), (188, 8), alto_four_musicmaker_one],
1706     [(188, 8), (190, 8), alto_four_musicmaker_one],
1707     [(190, 8), (192, 8), alto_four_musicmaker_one],
1708     [(192, 8), (194, 8), alto_four_musicmaker_one],
1709     [(194, 8), (196, 8), alto_four_musicmaker_two],
1710     [(196, 8), (198, 8), alto_four_musicmaker_two],
1711     [(198, 8), (199, 8), alto_four_musicmaker_one],
1712     [(199, 8), (200, 8), alto_four_musicmaker_one],
1713 ]
1714 ])
1715
1716 voice_9_timespan_list = abjad.TimespanList([
1717     abjad.AnnotatedTimespan(
1718         start_offset=start_offset,
1719         stop_offset=stop_offset,
1720         annotation=MusicSpecifier(
1721             music_maker=music_maker,
1722             voice_name='Voice 9',
1723         ),
1724     )
1725     for start_offset, stop_offset, music_maker in [
1726         [(0, 8), (2, 8), alto_five_musicmaker_one],
1727         [(2, 8), (4, 8), alto_five_musicmaker_one],
1728         [(4, 8), (6, 8), alto_five_musicmaker_one],
1729         [(6, 8), (8, 8), alto_five_musicmaker_two],
1730         [(8, 8), (10, 8), alto_five_musicmaker_two],
1731         [(10, 8), (12, 8), alto_five_musicmaker_one],
1732         [(12, 8), (14, 8), alto_five_musicmaker_one],
1733         [(14, 8), (16, 8), alto_five_musicmaker_two],
1734         [(16, 8), (18, 8), alto_five_musicmaker_one],
1735         [(18, 8), (20, 8), alto_five_musicmaker_one],
1736         [(20, 8), (22, 8), alto_five_musicmaker_two],
1737         [(22, 8), (24, 8), alto_five_musicmaker_two],
1738         [(24, 8), (26, 8), alto_five_musicmaker_one],
1739         [(26, 8), (28, 8), alto_five_musicmaker_two],
1740         [(28, 8), (30, 8), alto_five_musicmaker_one],
1741         [(30, 8), (32, 8), alto_five_musicmaker_two],
1742         [(32, 8), (34, 8), alto_five_musicmaker_one],
1743         [(34, 8), (36, 8), alto_five_musicmaker_one],
1744         [(36, 8), (38, 8), alto_five_musicmaker_one],
1745         [(38, 8), (40, 8), alto_five_musicmaker_two],
1746         [(40, 8), (42, 8), alto_five_musicmaker_two],
1747         [(42, 8), (44, 8), alto_five_musicmaker_one],
1748         [(44, 8), (46, 8), alto_five_musicmaker_one],
1749         [(46, 8), (48, 8), alto_five_musicmaker_one],
1750         [(48, 8), (50, 8), alto_five_musicmaker_two],
1751         [(50, 8), (52, 8), alto_five_musicmaker_two],
1752         [(52, 8), (54, 8), alto_five_musicmaker_two],
1753         [(54, 8), (56, 8), alto_five_musicmaker_one],
1754         [(56, 8), (58, 8), alto_five_musicmaker_one],
1755         [(58, 8), (60, 8), alto_five_musicmaker_one],

```

```
1756      [(60, 8), (62, 8), alto_five_musicmaker_two],  
1757      [(62, 8), (64, 8), alto_five_musicmaker_one],  
1758      [(64, 8), (66, 8), alto_five_musicmaker_one],  
1759      [(66, 8), (68, 8), alto_five_musicmaker_two],  
1760      [(68, 8), (70, 8), alto_five_musicmaker_one],  
1761      [(70, 8), (72, 8), alto_five_musicmaker_one],  
1762      [(72, 8), (74, 8), alto_five_musicmaker_two],  
1763      [(74, 8), (76, 8), alto_five_musicmaker_two],  
1764      [(76, 8), (78, 8), alto_five_musicmaker_two],  
1765      [(78, 8), (80, 8), alto_five_musicmaker_two],  
1766      [(80, 8), (82, 8), alto_five_musicmaker_one],  
1767      [(82, 8), (84, 8), alto_five_musicmaker_one],  
1768      [(84, 8), (86, 8), alto_five_musicmaker_one],  
1769      [(86, 8), (88, 8), alto_five_musicmaker_one],  
1770      [(88, 8), (90, 8), alto_five_musicmaker_two],  
1771      [(90, 8), (92, 8), alto_five_musicmaker_one],  
1772      [(92, 8), (94, 8), alto_five_musicmaker_one],  
1773      [(94, 8), (96, 8), alto_five_musicmaker_one],  
1774      [(96, 8), (98, 8), alto_five_musicmaker_two],  
1775      [(98, 8), (100, 8), alto_five_musicmaker_one],  
1776      [(100, 8), (102, 8), alto_five_musicmaker_two],  
1777      [(102, 8), (104, 8), alto_five_musicmaker_one],  
1778      [(104, 8), (106, 8), alto_five_musicmaker_one],  
1779      [(106, 8), (108, 8), alto_five_musicmaker_one],  
1780      [(108, 8), (110, 8), alto_five_musicmaker_two],  
1781      [(110, 8), (112, 8), alto_five_musicmaker_two],  
1782      [(112, 8), (114, 8), alto_five_musicmaker_two],  
1783      [(114, 8), (116, 8), alto_five_musicmaker_one],  
1784      [(116, 8), (118, 8), alto_five_musicmaker_one],  
1785      [(118, 8), (120, 8), alto_five_musicmaker_one],  
1786      [(120, 8), (122, 8), alto_five_musicmaker_one],  
1787      [(122, 8), (124, 8), alto_five_musicmaker_one],  
1788      [(124, 8), (126, 8), alto_five_musicmaker_one],  
1789      [(126, 8), (128, 8), alto_five_musicmaker_one],  
1790      [(128, 8), (130, 8), alto_five_musicmaker_one],  
1791      [(130, 8), (132, 8), alto_five_musicmaker_two],  
1792      [(132, 8), (134, 8), alto_five_musicmaker_two],  
1793      [(134, 8), (136, 8), alto_five_musicmaker_one],  
1794      [(136, 8), (138, 8), alto_five_musicmaker_one],  
1795      [(138, 8), (140, 8), alto_five_musicmaker_one],  
1796      [(140, 8), (142, 8), alto_five_musicmaker_two],  
1797      [(142, 8), (144, 8), alto_five_musicmaker_one],  
1798      [(144, 8), (146, 8), alto_five_musicmaker_two],  
1799      [(146, 8), (148, 8), alto_five_musicmaker_two],  
1800      [(148, 8), (150, 8), alto_five_musicmaker_one],  
1801      [(150, 8), (152, 8), alto_five_musicmaker_one],  
1802      [(152, 8), (154, 8), alto_five_musicmaker_one],  
1803      [(154, 8), (156, 8), alto_five_musicmaker_one],  
1804      [(156, 8), (158, 8), alto_five_musicmaker_one],  
1805      [(158, 8), (160, 8), alto_five_musicmaker_two],  
1806      [(160, 8), (162, 8), alto_five_musicmaker_two],  
1807      [(162, 8), (164, 8), alto_five_musicmaker_two],  
1808      [(164, 8), (166, 8), alto_five_musicmaker_one],  
1809      [(166, 8), (168, 8), alto_five_musicmaker_one],  
1810      [(168, 8), (170, 8), alto_five_musicmaker_one],  
1811      [(170, 8), (172, 8), alto_five_musicmaker_one],  
1812      [(172, 8), (174, 8), alto_five_musicmaker_one],  
1813      [(174, 8), (176, 8), alto_five_musicmaker_one],  
1814      [(176, 8), (178, 8), alto_five_musicmaker_two],
```

```

1815     [(178, 8), (180, 8), alto_five_musicmaker_one],
1816     [(180, 8), (182, 8), alto_five_musicmaker_two],
1817     [(182, 8), (184, 8), alto_five_musicmaker_two],
1818     [(184, 8), (186, 8), alto_five_musicmaker_one],
1819     [(186, 8), (188, 8), alto_five_musicmaker_one],
1820     [(188, 8), (190, 8), alto_five_musicmaker_one],
1821     [(190, 8), (192, 8), alto_five_musicmaker_one],
1822     [(192, 8), (194, 8), alto_five_musicmaker_one],
1823     [(194, 8), (196, 8), alto_five_musicmaker_one],
1824     [(196, 8), (198, 8), alto_five_musicmaker_two],
1825     [(198, 8), (199, 8), alto_five_musicmaker_one],
1826     [(199, 8), (200, 8), alto_five_musicmaker_two],
1827 ]
1828 ])
1829
1830 voice_10_timespan_list = abjad.TimespanList([
1831     abjad.AnnotatedTimespan(
1832         start_offset=start_offset,
1833         stop_offset=stop_offset,
1834         annotation=MusicSpecifier(
1835             music_maker=music_maker,
1836             voice_name='Voice 10',
1837         ),
1838     )
1839     for start_offset, stop_offset, music_maker in [
1840         [(0, 8), (2, 8), alto_six_musicmaker_one],
1841         [(2, 8), (4, 8), alto_six_musicmaker_one],
1842         [(4, 8), (6, 8), alto_six_musicmaker_one],
1843         [(6, 8), (8, 8), alto_six_musicmaker_one],
1844         [(8, 8), (10, 8), alto_six_musicmaker_two],
1845         [(10, 8), (12, 8), alto_six_musicmaker_two],
1846         [(12, 8), (14, 8), alto_six_musicmaker_two],
1847         [(14, 8), (16, 8), alto_six_musicmaker_one],
1848         [(16, 8), (18, 8), alto_six_musicmaker_one],
1849         [(18, 8), (20, 8), alto_six_musicmaker_one],
1850         [(20, 8), (22, 8), alto_six_musicmaker_one],
1851         [(22, 8), (24, 8), alto_six_musicmaker_one],
1852         [(24, 8), (26, 8), alto_six_musicmaker_one],
1853         [(26, 8), (28, 8), alto_six_musicmaker_two],
1854         [(28, 8), (30, 8), alto_six_musicmaker_two],
1855         [(30, 8), (32, 8), alto_six_musicmaker_two],
1856         [(32, 8), (34, 8), alto_six_musicmaker_one],
1857         [(34, 8), (36, 8), alto_six_musicmaker_one],
1858         [(36, 8), (38, 8), alto_six_musicmaker_two],
1859         [(38, 8), (40, 8), alto_six_musicmaker_two],
1860         [(40, 8), (42, 8), alto_six_musicmaker_one],
1861         [(42, 8), (44, 8), alto_six_musicmaker_one],
1862         [(44, 8), (46, 8), alto_six_musicmaker_one],
1863         [(46, 8), (48, 8), alto_six_musicmaker_one],
1864         [(48, 8), (50, 8), alto_six_musicmaker_one],
1865         [(50, 8), (52, 8), alto_six_musicmaker_two],
1866         [(52, 8), (54, 8), alto_six_musicmaker_one],
1867         [(54, 8), (56, 8), alto_six_musicmaker_one],
1868         [(56, 8), (58, 8), alto_six_musicmaker_one],
1869         [(58, 8), (60, 8), alto_six_musicmaker_one],
1870         [(60, 8), (62, 8), alto_six_musicmaker_one],
1871         [(62, 8), (64, 8), alto_six_musicmaker_two],
1872         [(64, 8), (66, 8), alto_six_musicmaker_two],
1873         [(66, 8), (68, 8), alto_six_musicmaker_two],

```

```
1874      [(68, 8), (70, 8), alto_six_musicmaker_one],  
1875      [(70, 8), (72, 8), alto_six_musicmaker_one],  
1876      [(72, 8), (74, 8), alto_six_musicmaker_two],  
1877      [(74, 8), (76, 8), alto_six_musicmaker_one],  
1878      [(76, 8), (78, 8), alto_six_musicmaker_one],  
1879      [(78, 8), (80, 8), alto_six_musicmaker_two],  
1880      [(80, 8), (82, 8), alto_six_musicmaker_two],  
1881      [(82, 8), (84, 8), alto_six_musicmaker_one],  
1882      [(84, 8), (86, 8), alto_six_musicmaker_one],  
1883      [(86, 8), (88, 8), alto_six_musicmaker_one],  
1884      [(88, 8), (90, 8), alto_six_musicmaker_two],  
1885      [(90, 8), (92, 8), alto_six_musicmaker_one],  
1886      [(92, 8), (94, 8), alto_six_musicmaker_one],  
1887      [(94, 8), (96, 8), alto_six_musicmaker_one],  
1888      [(96, 8), (98, 8), alto_six_musicmaker_two],  
1889      [(98, 8), (100, 8), alto_six_musicmaker_two],  
1890      [(100, 8), (102, 8), alto_six_musicmaker_one],  
1891      [(102, 8), (104, 8), alto_six_musicmaker_one],  
1892      [(104, 8), (106, 8), alto_six_musicmaker_two],  
1893      [(106, 8), (108, 8), alto_six_musicmaker_two],  
1894      [(108, 8), (110, 8), alto_six_musicmaker_one],  
1895      [(110, 8), (112, 8), alto_six_musicmaker_two],  
1896      [(112, 8), (114, 8), alto_six_musicmaker_two],  
1897      [(114, 8), (116, 8), alto_six_musicmaker_one],  
1898      [(116, 8), (118, 8), alto_six_musicmaker_one],  
1899      [(118, 8), (120, 8), alto_six_musicmaker_one],  
1900      [(120, 8), (122, 8), alto_six_musicmaker_one],  
1901      [(122, 8), (124, 8), alto_six_musicmaker_two],  
1902      [(124, 8), (126, 8), alto_six_musicmaker_two],  
1903      [(126, 8), (128, 8), alto_six_musicmaker_two],  
1904      [(128, 8), (130, 8), alto_six_musicmaker_one],  
1905      [(130, 8), (132, 8), alto_six_musicmaker_two],  
1906      [(132, 8), (134, 8), alto_six_musicmaker_one],  
1907      [(134, 8), (136, 8), alto_six_musicmaker_one],  
1908      [(136, 8), (138, 8), alto_six_musicmaker_one],  
1909      [(138, 8), (140, 8), alto_six_musicmaker_one],  
1910      [(140, 8), (142, 8), alto_six_musicmaker_one],  
1911      [(142, 8), (144, 8), alto_six_musicmaker_two],  
1912      [(144, 8), (146, 8), alto_six_musicmaker_one],  
1913      [(146, 8), (148, 8), alto_six_musicmaker_one],  
1914      [(148, 8), (150, 8), alto_six_musicmaker_two],  
1915      [(150, 8), (152, 8), alto_six_musicmaker_one],  
1916      [(152, 8), (154, 8), alto_six_musicmaker_one],  
1917      [(154, 8), (156, 8), alto_six_musicmaker_two],  
1918      [(156, 8), (158, 8), alto_six_musicmaker_one],  
1919      [(158, 8), (160, 8), alto_six_musicmaker_one],  
1920      [(160, 8), (162, 8), alto_six_musicmaker_one],  
1921      [(162, 8), (164, 8), alto_six_musicmaker_one],  
1922      [(164, 8), (166, 8), alto_six_musicmaker_two],  
1923      [(166, 8), (168, 8), alto_six_musicmaker_two],  
1924      [(168, 8), (170, 8), alto_six_musicmaker_one],  
1925      [(170, 8), (172, 8), alto_six_musicmaker_one],  
1926      [(172, 8), (174, 8), alto_six_musicmaker_two],  
1927      [(174, 8), (176, 8), alto_six_musicmaker_two],  
1928      [(176, 8), (178, 8), alto_six_musicmaker_two],  
1929      [(178, 8), (180, 8), alto_six_musicmaker_one],  
1930      [(180, 8), (182, 8), alto_six_musicmaker_one],  
1931      [(182, 8), (184, 8), alto_six_musicmaker_one],  
1932      [(184, 8), (186, 8), alto_six_musicmaker_one],
```

```

1933     [(186, 8), (188, 8), alto_six_musicmaker_one],
1934     [(188, 8), (190, 8), alto_six_musicmaker_two],
1935     [(190, 8), (192, 8), alto_six_musicmaker_one],
1936     [(192, 8), (194, 8), alto_six_musicmaker_one],
1937     [(194, 8), (196, 8), alto_six_musicmaker_one],
1938     [(196, 8), (198, 8), alto_six_musicmaker_two],
1939     [(198, 8), (199, 8), alto_six_musicmaker_one],
1940     [(199, 8), (200, 8), alto_six_musicmaker_one],
1941 ]
1942 ])
1943
1944 voice_11_timespan_list = abjad.TimespanList([
1945     abjad.AnnotatedTimespan(
1946         start_offset=start_offset,
1947         stop_offset=stop_offset,
1948         annotation=MusicSpecifier(
1949             music_maker=music_maker,
1950             voice_name='Voice 11',
1951         ),
1952     )
1953     for start_offset, stop_offset, music_maker in [
1954         [(0, 8), (2, 8), tenor_one_musicmaker_one],
1955         [(2, 8), (4, 8), tenor_one_musicmaker_one],
1956         [(4, 8), (6, 8), tenor_one_musicmaker_one],
1957         [(6, 8), (8, 8), tenor_one_musicmaker_two],
1958         [(8, 8), (10, 8), tenor_one_musicmaker_one],
1959         [(10, 8), (12, 8), tenor_one_musicmaker_one],
1960         [(12, 8), (14, 8), tenor_one_musicmaker_one],
1961         [(14, 8), (16, 8), tenor_one_musicmaker_one],
1962         [(16, 8), (18, 8), tenor_one_musicmaker_two],
1963         [(18, 8), (20, 8), tenor_one_musicmaker_two],
1964         [(20, 8), (22, 8), tenor_one_musicmaker_one],
1965         [(22, 8), (24, 8), tenor_one_musicmaker_one],
1966         [(24, 8), (26, 8), tenor_one_musicmaker_one],
1967         [(26, 8), (28, 8), tenor_one_musicmaker_one],
1968         [(28, 8), (30, 8), tenor_one_musicmaker_two],
1969         [(30, 8), (32, 8), tenor_one_musicmaker_one],
1970         [(32, 8), (34, 8), tenor_one_musicmaker_two],
1971         [(34, 8), (36, 8), tenor_one_musicmaker_one],
1972         [(36, 8), (38, 8), tenor_one_musicmaker_one],
1973         [(38, 8), (40, 8), tenor_one_musicmaker_one],
1974         [(40, 8), (42, 8), tenor_one_musicmaker_one],
1975         [(42, 8), (44, 8), tenor_one_musicmaker_one],
1976         [(44, 8), (46, 8), tenor_one_musicmaker_two],
1977         [(46, 8), (48, 8), tenor_one_musicmaker_one],
1978         [(48, 8), (50, 8), tenor_one_musicmaker_one],
1979         [(50, 8), (52, 8), tenor_one_musicmaker_two],
1980         [(52, 8), (54, 8), tenor_one_musicmaker_one],
1981         [(54, 8), (56, 8), tenor_one_musicmaker_one],
1982         [(56, 8), (58, 8), tenor_one_musicmaker_one],
1983         [(58, 8), (60, 8), tenor_one_musicmaker_one],
1984         [(60, 8), (62, 8), tenor_one_musicmaker_one],
1985         [(62, 8), (64, 8), tenor_one_musicmaker_two],
1986         [(64, 8), (66, 8), tenor_one_musicmaker_two],
1987         [(66, 8), (68, 8), tenor_one_musicmaker_two],
1988         [(68, 8), (70, 8), tenor_one_musicmaker_one],
1989         [(70, 8), (72, 8), tenor_one_musicmaker_one],
1990         [(72, 8), (74, 8), tenor_one_musicmaker_one],
1991         [(74, 8), (76, 8), tenor_one_musicmaker_one],

```

```

1992      [(76, 8), (78, 8), tenor_one_musicmaker_one],
1993      [(78, 8), (80, 8), tenor_one_musicmaker_one],
1994      [(80, 8), (82, 8), tenor_one_musicmaker_one],
1995      [(82, 8), (84, 8), tenor_one_musicmaker_one],
1996      [(84, 8), (86, 8), tenor_one_musicmaker_one],
1997      [(86, 8), (88, 8), tenor_one_musicmaker_one],
1998      [(88, 8), (90, 8), tenor_one_musicmaker_two],
1999      [(90, 8), (92, 8), tenor_one_musicmaker_two],
2000      [(92, 8), (94, 8), tenor_one_musicmaker_one],
2001      [(94, 8), (96, 8), tenor_one_musicmaker_one],
2002      [(96, 8), (98, 8), tenor_one_musicmaker_one],
2003      [(98, 8), (100, 8), tenor_one_musicmaker_one],
2004      [(100, 8), (102, 8), tenor_one_musicmaker_two],
2005      [(102, 8), (104, 8), tenor_one_musicmaker_two],
2006      [(104, 8), (106, 8), tenor_one_musicmaker_two],
2007      [(106, 8), (108, 8), tenor_one_musicmaker_one],
2008      [(108, 8), (110, 8), tenor_one_musicmaker_one],
2009      [(110, 8), (112, 8), tenor_one_musicmaker_one],
2010      [(112, 8), (114, 8), tenor_one_musicmaker_one],
2011      [(114, 8), (116, 8), tenor_one_musicmaker_one],
2012      [(116, 8), (118, 8), tenor_one_musicmaker_one],
2013      [(118, 8), (120, 8), tenor_one_musicmaker_one],
2014      [(120, 8), (122, 8), tenor_one_musicmaker_two],
2015      [(122, 8), (124, 8), tenor_one_musicmaker_two],
2016      [(124, 8), (126, 8), tenor_one_musicmaker_two],
2017      [(126, 8), (128, 8), tenor_one_musicmaker_one],
2018      [(128, 8), (130, 8), tenor_one_musicmaker_one],
2019      [(130, 8), (132, 8), tenor_one_musicmaker_one],
2020      [(132, 8), (134, 8), tenor_one_musicmaker_two],
2021      [(134, 8), (136, 8), tenor_one_musicmaker_two],
2022      [(136, 8), (138, 8), tenor_one_musicmaker_two],
2023      [(138, 8), (140, 8), tenor_one_musicmaker_one],
2024      [(140, 8), (142, 8), tenor_one_musicmaker_one],
2025      [(142, 8), (144, 8), tenor_one_musicmaker_one],
2026      [(144, 8), (146, 8), tenor_one_musicmaker_one],
2027      [(146, 8), (148, 8), tenor_one_musicmaker_two],
2028      [(148, 8), (150, 8), tenor_one_musicmaker_one],
2029      [(150, 8), (152, 8), tenor_one_musicmaker_one],
2030      [(152, 8), (154, 8), tenor_one_musicmaker_two],
2031      [(154, 8), (156, 8), tenor_one_musicmaker_two],
2032      [(156, 8), (158, 8), tenor_one_musicmaker_one],
2033      [(158, 8), (160, 8), tenor_one_musicmaker_two],
2034      [(160, 8), (162, 8), tenor_one_musicmaker_two],
2035      [(162, 8), (164, 8), tenor_one_musicmaker_one],
2036      [(164, 8), (166, 8), tenor_one_musicmaker_one],
2037      [(166, 8), (168, 8), tenor_one_musicmaker_one],
2038      [(168, 8), (170, 8), tenor_one_musicmaker_one],
2039      [(170, 8), (172, 8), tenor_one_musicmaker_one],
2040      [(172, 8), (174, 8), tenor_one_musicmaker_two],
2041      [(174, 8), (176, 8), tenor_one_musicmaker_one],
2042      [(176, 8), (178, 8), tenor_one_musicmaker_one],
2043      [(178, 8), (180, 8), tenor_one_musicmaker_two],
2044      [(180, 8), (182, 8), tenor_one_musicmaker_two],
2045      [(182, 8), (184, 8), tenor_one_musicmaker_one],
2046      [(184, 8), (186, 8), tenor_one_musicmaker_one],
2047      [(186, 8), (188, 8), tenor_one_musicmaker_one],
2048      [(188, 8), (190, 8), tenor_one_musicmaker_one],
2049      [(190, 8), (192, 8), tenor_one_musicmaker_two],
2050      [(192, 8), (194, 8), tenor_one_musicmaker_two],

```

```
2051     [(194, 8), (196, 8), tenor_one_musicmaker_one],
2052     [(196, 8), (198, 8), tenor_one_musicmaker_one],
2053     [(198, 8), (199, 8), tenor_one_musicmaker_one],
2054     [(199, 8), (200, 8), tenor_one_musicmaker_one],
2055 ]
2056 ])
2057
2058 voice_12_timespan_list = abjad.TimespanList([
2059     abjad.AnnotatedTimespan(
2060         start_offset=start_offset,
2061         stop_offset=stop_offset,
2062         annotation=MusicSpecifier(
2063             music_maker=music_maker,
2064             voice_name='Voice 12',
2065         ),
2066     )
2067     for start_offset, stop_offset, music_maker in [
2068         [(0, 8), (2, 8), tenor_two_musicmaker_one],
2069         [(2, 8), (4, 8), tenor_two_musicmaker_two],
2070         [(4, 8), (6, 8), tenor_two_musicmaker_two],
2071         [(6, 8), (8, 8), tenor_two_musicmaker_one],
2072         [(8, 8), (10, 8), tenor_two_musicmaker_one],
2073         [(10, 8), (12, 8), tenor_two_musicmaker_one],
2074         [(12, 8), (14, 8), tenor_two_musicmaker_one],
2075         [(14, 8), (16, 8), tenor_two_musicmaker_one],
2076         [(16, 8), (18, 8), tenor_two_musicmaker_one],
2077         [(18, 8), (20, 8), tenor_two_musicmaker_two],
2078         [(20, 8), (22, 8), tenor_two_musicmaker_two],
2079         [(22, 8), (24, 8), tenor_two_musicmaker_two],
2080         [(24, 8), (26, 8), tenor_two_musicmaker_one],
2081         [(26, 8), (28, 8), tenor_two_musicmaker_one],
2082         [(28, 8), (30, 8), tenor_two_musicmaker_one],
2083         [(30, 8), (32, 8), tenor_two_musicmaker_one],
2084         [(32, 8), (34, 8), tenor_two_musicmaker_one],
2085         [(34, 8), (36, 8), tenor_two_musicmaker_one],
2086         [(36, 8), (38, 8), tenor_two_musicmaker_two],
2087         [(38, 8), (40, 8), tenor_two_musicmaker_two],
2088         [(40, 8), (42, 8), tenor_two_musicmaker_two],
2089         [(42, 8), (44, 8), tenor_two_musicmaker_one],
2090         [(44, 8), (46, 8), tenor_two_musicmaker_one],
2091         [(46, 8), (48, 8), tenor_two_musicmaker_one],
2092         [(48, 8), (50, 8), tenor_two_musicmaker_one],
2093         [(50, 8), (52, 8), tenor_two_musicmaker_two],
2094         [(52, 8), (54, 8), tenor_two_musicmaker_two],
2095         [(54, 8), (56, 8), tenor_two_musicmaker_one],
2096         [(56, 8), (58, 8), tenor_two_musicmaker_one],
2097         [(58, 8), (60, 8), tenor_two_musicmaker_one],
2098         [(60, 8), (62, 8), tenor_two_musicmaker_one],
2099         [(62, 8), (64, 8), tenor_two_musicmaker_one],
2100         [(64, 8), (66, 8), tenor_two_musicmaker_two],
2101         [(66, 8), (68, 8), tenor_two_musicmaker_two],
2102         [(68, 8), (70, 8), tenor_two_musicmaker_two],
2103         [(70, 8), (72, 8), tenor_two_musicmaker_one],
2104         [(72, 8), (74, 8), tenor_two_musicmaker_one],
2105         [(74, 8), (76, 8), tenor_two_musicmaker_one],
2106         [(76, 8), (78, 8), tenor_two_musicmaker_one],
2107         [(78, 8), (80, 8), tenor_two_musicmaker_one],
2108         [(80, 8), (82, 8), tenor_two_musicmaker_one],
2109         [(82, 8), (84, 8), tenor_two_musicmaker_two],
```

```
2110      [(84, 8), (86, 8), tenor_two_musicmaker_two],  
2111      [(86, 8), (88, 8), tenor_two_musicmaker_two],  
2112      [(88, 8), (90, 8), tenor_two_musicmaker_one],  
2113      [(90, 8), (92, 8), tenor_two_musicmaker_one],  
2114      [(92, 8), (94, 8), tenor_two_musicmaker_one],  
2115      [(94, 8), (96, 8), tenor_two_musicmaker_one],  
2116      [(96, 8), (98, 8), tenor_two_musicmaker_one],  
2117      [(98, 8), (100, 8), tenor_two_musicmaker_two],  
2118      [(100, 8), (102, 8), tenor_two_musicmaker_two],  
2119      [(102, 8), (104, 8), tenor_two_musicmaker_two],  
2120      [(104, 8), (106, 8), tenor_two_musicmaker_one],  
2121      [(106, 8), (108, 8), tenor_two_musicmaker_one],  
2122      [(108, 8), (110, 8), tenor_two_musicmaker_one],  
2123      [(110, 8), (112, 8), tenor_two_musicmaker_one],  
2124      [(112, 8), (114, 8), tenor_two_musicmaker_two],  
2125      [(114, 8), (116, 8), tenor_two_musicmaker_two],  
2126      [(116, 8), (118, 8), tenor_two_musicmaker_two],  
2127      [(118, 8), (120, 8), tenor_two_musicmaker_one],  
2128      [(120, 8), (122, 8), tenor_two_musicmaker_one],  
2129      [(122, 8), (124, 8), tenor_two_musicmaker_one],  
2130      [(124, 8), (126, 8), tenor_two_musicmaker_one],  
2131      [(126, 8), (128, 8), tenor_two_musicmaker_one],  
2132      [(128, 8), (130, 8), tenor_two_musicmaker_one],  
2133      [(130, 8), (132, 8), tenor_two_musicmaker_one],  
2134      [(132, 8), (134, 8), tenor_two_musicmaker_two],  
2135      [(134, 8), (136, 8), tenor_two_musicmaker_two],  
2136      [(136, 8), (138, 8), tenor_two_musicmaker_two],  
2137      [(138, 8), (140, 8), tenor_two_musicmaker_one],  
2138      [(140, 8), (142, 8), tenor_two_musicmaker_one],  
2139      [(142, 8), (144, 8), tenor_two_musicmaker_one],  
2140      [(144, 8), (146, 8), tenor_two_musicmaker_one],  
2141      [(146, 8), (148, 8), tenor_two_musicmaker_one],  
2142      [(148, 8), (150, 8), tenor_two_musicmaker_two],  
2143      [(150, 8), (152, 8), tenor_two_musicmaker_two],  
2144      [(152, 8), (154, 8), tenor_two_musicmaker_one],  
2145      [(154, 8), (156, 8), tenor_two_musicmaker_one],  
2146      [(156, 8), (158, 8), tenor_two_musicmaker_two],  
2147      [(158, 8), (160, 8), tenor_two_musicmaker_one],  
2148      [(160, 8), (162, 8), tenor_two_musicmaker_one],  
2149      [(162, 8), (164, 8), tenor_two_musicmaker_one],  
2150      [(164, 8), (166, 8), tenor_two_musicmaker_two],  
2151      [(166, 8), (168, 8), tenor_two_musicmaker_one],  
2152      [(168, 8), (170, 8), tenor_two_musicmaker_two],  
2153      [(170, 8), (172, 8), tenor_two_musicmaker_two],  
2154      [(172, 8), (174, 8), tenor_two_musicmaker_one],  
2155      [(174, 8), (176, 8), tenor_two_musicmaker_one],  
2156      [(176, 8), (178, 8), tenor_two_musicmaker_one],  
2157      [(178, 8), (180, 8), tenor_two_musicmaker_two],  
2158      [(180, 8), (182, 8), tenor_two_musicmaker_two],  
2159      [(182, 8), (184, 8), tenor_two_musicmaker_two],  
2160      [(184, 8), (186, 8), tenor_two_musicmaker_two],  
2161      [(186, 8), (188, 8), tenor_two_musicmaker_one],  
2162      [(188, 8), (190, 8), tenor_two_musicmaker_one],  
2163      [(190, 8), (192, 8), tenor_two_musicmaker_one],  
2164      [(192, 8), (194, 8), tenor_two_musicmaker_one],  
2165      [(194, 8), (196, 8), tenor_two_musicmaker_two],  
2166      [(196, 8), (198, 8), tenor_two_musicmaker_one],  
2167      [(198, 8), (199, 8), tenor_two_musicmaker_two],  
2168      [(199, 8), (200, 8), tenor_two_musicmaker_two],
```

```
2169     ]
2170 ])
2171
2172 voice_13_timespan_list = abjad.TimespanList([
2173     abjad.AnnotatedTimespan(
2174         start_offset=start_offset,
2175         stop_offset=stop_offset,
2176         annotation=MusicSpecifier(
2177             music_maker=music_maker,
2178             voice_name='Voice 13',
2179         ),
2180     )
2181     for start_offset, stop_offset, music_maker in [
2182         [(0, 8), (2, 8), tenor_three_musicmaker_one],
2183         [(2, 8), (4, 8), tenor_three_musicmaker_one],
2184         [(4, 8), (6, 8), tenor_three_musicmaker_two],
2185         [(6, 8), (8, 8), tenor_three_musicmaker_two],
2186         [(8, 8), (10, 8), tenor_three_musicmaker_one],
2187         [(10, 8), (12, 8), tenor_three_musicmaker_two],
2188         [(12, 8), (14, 8), tenor_three_musicmaker_two],
2189         [(14, 8), (16, 8), tenor_three_musicmaker_one],
2190         [(16, 8), (18, 8), tenor_three_musicmaker_one],
2191         [(18, 8), (20, 8), tenor_three_musicmaker_one],
2192         [(20, 8), (22, 8), tenor_three_musicmaker_one],
2193         [(22, 8), (24, 8), tenor_three_musicmaker_one],
2194         [(24, 8), (26, 8), tenor_three_musicmaker_two],
2195         [(26, 8), (28, 8), tenor_three_musicmaker_two],
2196         [(28, 8), (30, 8), tenor_three_musicmaker_two],
2197         [(30, 8), (32, 8), tenor_three_musicmaker_one],
2198         [(32, 8), (34, 8), tenor_three_musicmaker_one],
2199         [(34, 8), (36, 8), tenor_three_musicmaker_one],
2200         [(36, 8), (38, 8), tenor_three_musicmaker_one],
2201         [(38, 8), (40, 8), tenor_three_musicmaker_two],
2202         [(40, 8), (42, 8), tenor_three_musicmaker_two],
2203         [(42, 8), (44, 8), tenor_three_musicmaker_two],
2204         [(44, 8), (46, 8), tenor_three_musicmaker_one],
2205         [(46, 8), (48, 8), tenor_three_musicmaker_one],
2206         [(48, 8), (50, 8), tenor_three_musicmaker_one],
2207         [(50, 8), (52, 8), tenor_three_musicmaker_one],
2208         [(52, 8), (54, 8), tenor_three_musicmaker_one],
2209         [(54, 8), (56, 8), tenor_three_musicmaker_two],
2210         [(56, 8), (58, 8), tenor_three_musicmaker_one],
2211         [(58, 8), (60, 8), tenor_three_musicmaker_one],
2212         [(60, 8), (62, 8), tenor_three_musicmaker_one],
2213         [(62, 8), (64, 8), tenor_three_musicmaker_one],
2214         [(64, 8), (66, 8), tenor_three_musicmaker_two],
2215         [(66, 8), (68, 8), tenor_three_musicmaker_two],
2216         [(68, 8), (70, 8), tenor_three_musicmaker_one],
2217         [(70, 8), (72, 8), tenor_three_musicmaker_one],
2218         [(72, 8), (74, 8), tenor_three_musicmaker_one],
2219         [(74, 8), (76, 8), tenor_three_musicmaker_one],
2220         [(76, 8), (78, 8), tenor_three_musicmaker_two],
2221         [(78, 8), (80, 8), tenor_three_musicmaker_two],
2222         [(80, 8), (82, 8), tenor_three_musicmaker_one],
2223         [(82, 8), (84, 8), tenor_three_musicmaker_one],
2224         [(84, 8), (86, 8), tenor_three_musicmaker_one],
2225         [(86, 8), (88, 8), tenor_three_musicmaker_two],
2226         [(88, 8), (90, 8), tenor_three_musicmaker_one],
2227         [(90, 8), (92, 8), tenor_three_musicmaker_two],
```

```

2228     [(92, 8), (94, 8), tenor_three_musicmaker_two],
2229     [(94, 8), (96, 8), tenor_three_musicmaker_one],
2230     [(96, 8), (98, 8), tenor_three_musicmaker_one],
2231     [(98, 8), (100, 8), tenor_three_musicmaker_two],
2232     [(100, 8), (102, 8), tenor_three_musicmaker_two],
2233     [(102, 8), (104, 8), tenor_three_musicmaker_one],
2234     [(104, 8), (106, 8), tenor_three_musicmaker_two],
2235     [(106, 8), (108, 8), tenor_three_musicmaker_one],
2236     [(108, 8), (110, 8), tenor_three_musicmaker_one],
2237     [(110, 8), (112, 8), tenor_three_musicmaker_one],
2238     [(112, 8), (114, 8), tenor_three_musicmaker_one],
2239     [(114, 8), (116, 8), tenor_three_musicmaker_one],
2240     [(116, 8), (118, 8), tenor_three_musicmaker_two],
2241     [(118, 8), (120, 8), tenor_three_musicmaker_two],
2242     [(120, 8), (122, 8), tenor_three_musicmaker_one],
2243     [(122, 8), (124, 8), tenor_three_musicmaker_one],
2244     [(124, 8), (126, 8), tenor_three_musicmaker_one],
2245     [(126, 8), (128, 8), tenor_three_musicmaker_two],
2246     [(128, 8), (130, 8), tenor_three_musicmaker_two],
2247     [(130, 8), (132, 8), tenor_three_musicmaker_one],
2248     [(132, 8), (134, 8), tenor_three_musicmaker_one],
2249     [(134, 8), (136, 8), tenor_three_musicmaker_one],
2250     [(136, 8), (138, 8), tenor_three_musicmaker_two],
2251     [(138, 8), (140, 8), tenor_three_musicmaker_one],
2252     [(140, 8), (142, 8), tenor_three_musicmaker_one],
2253     [(142, 8), (144, 8), tenor_three_musicmaker_one],
2254     [(144, 8), (146, 8), tenor_three_musicmaker_two],
2255     [(146, 8), (148, 8), tenor_three_musicmaker_two],
2256     [(148, 8), (150, 8), tenor_three_musicmaker_two],
2257     [(150, 8), (152, 8), tenor_three_musicmaker_one],
2258     [(152, 8), (154, 8), tenor_three_musicmaker_one],
2259     [(154, 8), (156, 8), tenor_three_musicmaker_one],
2260     [(156, 8), (158, 8), tenor_three_musicmaker_one],
2261     [(158, 8), (160, 8), tenor_three_musicmaker_two],
2262     [(160, 8), (162, 8), tenor_three_musicmaker_two],
2263     [(162, 8), (164, 8), tenor_three_musicmaker_one],
2264     [(164, 8), (166, 8), tenor_three_musicmaker_one],
2265     [(166, 8), (168, 8), tenor_three_musicmaker_one],
2266     [(168, 8), (170, 8), tenor_three_musicmaker_one],
2267     [(170, 8), (172, 8), tenor_three_musicmaker_two],
2268     [(172, 8), (174, 8), tenor_three_musicmaker_two],
2269     [(174, 8), (176, 8), tenor_three_musicmaker_two],
2270     [(176, 8), (178, 8), tenor_three_musicmaker_two],
2271     [(178, 8), (180, 8), tenor_three_musicmaker_one],
2272     [(180, 8), (182, 8), tenor_three_musicmaker_one],
2273     [(182, 8), (184, 8), tenor_three_musicmaker_one],
2274     [(184, 8), (186, 8), tenor_three_musicmaker_one],
2275     [(186, 8), (188, 8), tenor_three_musicmaker_two],
2276     [(188, 8), (190, 8), tenor_three_musicmaker_one],
2277     [(190, 8), (192, 8), tenor_three_musicmaker_one],
2278     [(192, 8), (194, 8), tenor_three_musicmaker_two],
2279     [(194, 8), (196, 8), tenor_three_musicmaker_one],
2280     [(196, 8), (198, 8), tenor_three_musicmaker_two],
2281     [(198, 8), (199, 8), tenor_three_musicmaker_one],
2282     [(199, 8), (200, 8), tenor_three_musicmaker_one],
2283 ]
2284 ])
2285
2286 voice_14_timespan_list = abjad.TimespanList([

```

```

2287 abjad.AnnotatedTimespan(
2288     start_offset=start_offset,
2289     stop_offset=stop_offset,
2290     annotation=MusicSpecifier(
2291         music_maker=music_maker,
2292         voice_name='Voice 14',
2293     ),
2294 )
2295 for start_offset, stop_offset, music_maker in [
2296     [(0, 8), (2, 8), tenor_four_musicmaker_one],
2297     [(2, 8), (4, 8), tenor_four_musicmaker_two],
2298     [(4, 8), (6, 8), tenor_four_musicmaker_two],
2299     [(6, 8), (8, 8), tenor_four_musicmaker_one],
2300     [(8, 8), (10, 8), tenor_four_musicmaker_one],
2301     [(10, 8), (12, 8), tenor_four_musicmaker_one],
2302     [(12, 8), (14, 8), tenor_four_musicmaker_one],
2303     [(14, 8), (16, 8), tenor_four_musicmaker_one],
2304     [(16, 8), (18, 8), tenor_four_musicmaker_one],
2305     [(18, 8), (20, 8), tenor_four_musicmaker_two],
2306     [(20, 8), (22, 8), tenor_four_musicmaker_two],
2307     [(22, 8), (24, 8), tenor_four_musicmaker_two],
2308     [(24, 8), (26, 8), tenor_four_musicmaker_one],
2309     [(26, 8), (28, 8), tenor_four_musicmaker_one],
2310     [(28, 8), (30, 8), tenor_four_musicmaker_one],
2311     [(30, 8), (32, 8), tenor_four_musicmaker_one],
2312     [(32, 8), (34, 8), tenor_four_musicmaker_two],
2313     [(34, 8), (36, 8), tenor_four_musicmaker_two],
2314     [(36, 8), (38, 8), tenor_four_musicmaker_one],
2315     [(38, 8), (40, 8), tenor_four_musicmaker_one],
2316     [(40, 8), (42, 8), tenor_four_musicmaker_one],
2317     [(42, 8), (44, 8), tenor_four_musicmaker_two],
2318     [(44, 8), (46, 8), tenor_four_musicmaker_two],
2319     [(46, 8), (48, 8), tenor_four_musicmaker_one],
2320     [(48, 8), (50, 8), tenor_four_musicmaker_one],
2321     [(50, 8), (52, 8), tenor_four_musicmaker_one],
2322     [(52, 8), (54, 8), tenor_four_musicmaker_one],
2323     [(54, 8), (56, 8), tenor_four_musicmaker_two],
2324     [(56, 8), (58, 8), tenor_four_musicmaker_two],
2325     [(58, 8), (60, 8), tenor_four_musicmaker_two],
2326     [(60, 8), (62, 8), tenor_four_musicmaker_one],
2327     [(62, 8), (64, 8), tenor_four_musicmaker_one],
2328     [(64, 8), (66, 8), tenor_four_musicmaker_one],
2329     [(66, 8), (68, 8), tenor_four_musicmaker_one],
2330     [(68, 8), (70, 8), tenor_four_musicmaker_two],
2331     [(70, 8), (72, 8), tenor_four_musicmaker_two],
2332     [(72, 8), (74, 8), tenor_four_musicmaker_two],
2333     [(74, 8), (76, 8), tenor_four_musicmaker_one],
2334     [(76, 8), (78, 8), tenor_four_musicmaker_one],
2335     [(78, 8), (80, 8), tenor_four_musicmaker_one],
2336     [(80, 8), (82, 8), tenor_four_musicmaker_two],
2337     [(82, 8), (84, 8), tenor_four_musicmaker_one],
2338     [(84, 8), (86, 8), tenor_four_musicmaker_one],
2339     [(86, 8), (88, 8), tenor_four_musicmaker_two],
2340     [(88, 8), (90, 8), tenor_four_musicmaker_one],
2341     [(90, 8), (92, 8), tenor_four_musicmaker_two],
2342     [(92, 8), (94, 8), tenor_four_musicmaker_one],
2343     [(94, 8), (96, 8), tenor_four_musicmaker_one],
2344     [(96, 8), (98, 8), tenor_four_musicmaker_one],
2345     [(98, 8), (100, 8), tenor_four_musicmaker_two],

```

```

2346     [(100, 8), (102, 8), tenor_four_musicmaker_two],
2347     [(102, 8), (104, 8), tenor_four_musicmaker_one],
2348     [(104, 8), (106, 8), tenor_four_musicmaker_one],
2349     [(106, 8), (108, 8), tenor_four_musicmaker_one],
2350     [(108, 8), (110, 8), tenor_four_musicmaker_two],
2351     [(110, 8), (112, 8), tenor_four_musicmaker_one],
2352     [(112, 8), (114, 8), tenor_four_musicmaker_one],
2353     [(114, 8), (116, 8), tenor_four_musicmaker_two],
2354     [(116, 8), (118, 8), tenor_four_musicmaker_one],
2355     [(118, 8), (120, 8), tenor_four_musicmaker_one],
2356     [(120, 8), (122, 8), tenor_four_musicmaker_two],
2357     [(122, 8), (124, 8), tenor_four_musicmaker_two],
2358     [(124, 8), (126, 8), tenor_four_musicmaker_one],
2359     [(126, 8), (128, 8), tenor_four_musicmaker_one],
2360     [(128, 8), (130, 8), tenor_four_musicmaker_two],
2361     [(130, 8), (132, 8), tenor_four_musicmaker_one],
2362     [(132, 8), (134, 8), tenor_four_musicmaker_two],
2363     [(134, 8), (136, 8), tenor_four_musicmaker_one],
2364     [(136, 8), (138, 8), tenor_four_musicmaker_two],
2365     [(138, 8), (140, 8), tenor_four_musicmaker_one],
2366     [(140, 8), (142, 8), tenor_four_musicmaker_one],
2367     [(142, 8), (144, 8), tenor_four_musicmaker_one],
2368     [(144, 8), (146, 8), tenor_four_musicmaker_one],
2369     [(146, 8), (148, 8), tenor_four_musicmaker_two],
2370     [(148, 8), (150, 8), tenor_four_musicmaker_one],
2371     [(150, 8), (152, 8), tenor_four_musicmaker_two],
2372     [(152, 8), (154, 8), tenor_four_musicmaker_one],
2373     [(154, 8), (156, 8), tenor_four_musicmaker_one],
2374     [(156, 8), (158, 8), tenor_four_musicmaker_one],
2375     [(158, 8), (160, 8), tenor_four_musicmaker_two],
2376     [(160, 8), (162, 8), tenor_four_musicmaker_two],
2377     [(162, 8), (164, 8), tenor_four_musicmaker_one],
2378     [(164, 8), (166, 8), tenor_four_musicmaker_one],
2379     [(166, 8), (168, 8), tenor_four_musicmaker_one],
2380     [(168, 8), (170, 8), tenor_four_musicmaker_one],
2381     [(170, 8), (172, 8), tenor_four_musicmaker_two],
2382     [(172, 8), (174, 8), tenor_four_musicmaker_one],
2383     [(174, 8), (176, 8), tenor_four_musicmaker_two],
2384     [(176, 8), (178, 8), tenor_four_musicmaker_one],
2385     [(178, 8), (180, 8), tenor_four_musicmaker_two],
2386     [(180, 8), (182, 8), tenor_four_musicmaker_two],
2387     [(182, 8), (184, 8), tenor_four_musicmaker_two],
2388     [(184, 8), (186, 8), tenor_four_musicmaker_one],
2389     [(186, 8), (188, 8), tenor_four_musicmaker_two],
2390     [(188, 8), (190, 8), tenor_four_musicmaker_one],
2391     [(190, 8), (192, 8), tenor_four_musicmaker_one],
2392     [(192, 8), (194, 8), tenor_four_musicmaker_one],
2393     [(194, 8), (196, 8), tenor_four_musicmaker_one],
2394     [(196, 8), (198, 8), tenor_four_musicmaker_two],
2395     [(198, 8), (199, 8), tenor_four_musicmaker_two],
2396     [(199, 8), (200, 8), tenor_four_musicmaker_one],
2397   ]
2398 ])
2399
2400 voice_15_timespan_list = abjad.TimespanList([
2401     abjad.AnnotatedTimespan(
2402         start_offset=start_offset,
2403         stop_offset=stop_offset,
2404         annotation=MusicSpecifier(

```

```
2405     music_maker=music_maker,
2406     voice_name='Voice 15',
2407   ),
2408 )
2409 for start_offset, stop_offset, music_maker in [
2410   [(0, 8), (2, 8), tenor_five_musicmaker_two],
2411   [(2, 8), (4, 8), tenor_five_musicmaker_two],
2412   [(4, 8), (6, 8), tenor_five_musicmaker_one],
2413   [(6, 8), (8, 8), tenor_five_musicmaker_one],
2414   [(8, 8), (10, 8), tenor_five_musicmaker_one],
2415   [(10, 8), (12, 8), tenor_five_musicmaker_two],
2416   [(12, 8), (14, 8), tenor_five_musicmaker_one],
2417   [(14, 8), (16, 8), tenor_five_musicmaker_one],
2418   [(16, 8), (18, 8), tenor_five_musicmaker_two],
2419   [(18, 8), (20, 8), tenor_five_musicmaker_two],
2420   [(20, 8), (22, 8), tenor_five_musicmaker_one],
2421   [(22, 8), (24, 8), tenor_five_musicmaker_one],
2422   [(24, 8), (26, 8), tenor_five_musicmaker_one],
2423   [(26, 8), (28, 8), tenor_five_musicmaker_one],
2424   [(28, 8), (30, 8), tenor_five_musicmaker_two],
2425   [(30, 8), (32, 8), tenor_five_musicmaker_two],
2426   [(32, 8), (34, 8), tenor_five_musicmaker_one],
2427   [(34, 8), (36, 8), tenor_five_musicmaker_one],
2428   [(36, 8), (38, 8), tenor_five_musicmaker_two],
2429   [(38, 8), (40, 8), tenor_five_musicmaker_two],
2430   [(40, 8), (42, 8), tenor_five_musicmaker_two],
2431   [(42, 8), (44, 8), tenor_five_musicmaker_one],
2432   [(44, 8), (46, 8), tenor_five_musicmaker_one],
2433   [(46, 8), (48, 8), tenor_five_musicmaker_one],
2434   [(48, 8), (50, 8), tenor_five_musicmaker_one],
2435   [(50, 8), (52, 8), tenor_five_musicmaker_one],
2436   [(52, 8), (54, 8), tenor_five_musicmaker_two],
2437   [(54, 8), (56, 8), tenor_five_musicmaker_two],
2438   [(56, 8), (58, 8), tenor_five_musicmaker_one],
2439   [(58, 8), (60, 8), tenor_five_musicmaker_one],
2440   [(60, 8), (62, 8), tenor_five_musicmaker_two],
2441   [(62, 8), (64, 8), tenor_five_musicmaker_two],
2442   [(64, 8), (66, 8), tenor_five_musicmaker_one],
2443   [(66, 8), (68, 8), tenor_five_musicmaker_one],
2444   [(68, 8), (70, 8), tenor_five_musicmaker_one],
2445   [(70, 8), (72, 8), tenor_five_musicmaker_one],
2446   [(72, 8), (74, 8), tenor_five_musicmaker_two],
2447   [(74, 8), (76, 8), tenor_five_musicmaker_two],
2448   [(76, 8), (78, 8), tenor_five_musicmaker_one],
2449   [(78, 8), (80, 8), tenor_five_musicmaker_one],
2450   [(80, 8), (82, 8), tenor_five_musicmaker_one],
2451   [(82, 8), (84, 8), tenor_five_musicmaker_one],
2452   [(84, 8), (86, 8), tenor_five_musicmaker_two],
2453   [(86, 8), (88, 8), tenor_five_musicmaker_one],
2454   [(88, 8), (90, 8), tenor_five_musicmaker_one],
2455   [(90, 8), (92, 8), tenor_five_musicmaker_two],
2456   [(92, 8), (94, 8), tenor_five_musicmaker_two],
2457   [(94, 8), (96, 8), tenor_five_musicmaker_one],
2458   [(96, 8), (98, 8), tenor_five_musicmaker_one],
2459   [(98, 8), (100, 8), tenor_five_musicmaker_one],
2460   [(100, 8), (102, 8), tenor_five_musicmaker_one],
2461   [(102, 8), (104, 8), tenor_five_musicmaker_two],
2462   [(104, 8), (106, 8), tenor_five_musicmaker_two],
2463   [(106, 8), (108, 8), tenor_five_musicmaker_one],
```

```

2464     [(108, 8), (110, 8), tenor_five_musicmaker_two],
2465     [(110, 8), (112, 8), tenor_five_musicmaker_one],
2466     [(112, 8), (114, 8), tenor_five_musicmaker_two],
2467     [(114, 8), (116, 8), tenor_five_musicmaker_two],
2468     [(116, 8), (118, 8), tenor_five_musicmaker_one],
2469     [(118, 8), (120, 8), tenor_five_musicmaker_two],
2470     [(120, 8), (122, 8), tenor_five_musicmaker_one],
2471     [(122, 8), (124, 8), tenor_five_musicmaker_two],
2472     [(124, 8), (126, 8), tenor_five_musicmaker_one],
2473     [(126, 8), (128, 8), tenor_five_musicmaker_one],
2474     [(128, 8), (130, 8), tenor_five_musicmaker_one],
2475     [(130, 8), (132, 8), tenor_five_musicmaker_one],
2476     [(132, 8), (134, 8), tenor_five_musicmaker_two],
2477     [(134, 8), (136, 8), tenor_five_musicmaker_two],
2478     [(136, 8), (138, 8), tenor_five_musicmaker_one],
2479     [(138, 8), (140, 8), tenor_five_musicmaker_one],
2480     [(140, 8), (142, 8), tenor_five_musicmaker_one],
2481     [(142, 8), (144, 8), tenor_five_musicmaker_two],
2482     [(144, 8), (146, 8), tenor_five_musicmaker_two],
2483     [(146, 8), (148, 8), tenor_five_musicmaker_two],
2484     [(148, 8), (150, 8), tenor_five_musicmaker_one],
2485     [(150, 8), (152, 8), tenor_five_musicmaker_one],
2486     [(152, 8), (154, 8), tenor_five_musicmaker_two],
2487     [(154, 8), (156, 8), tenor_five_musicmaker_one],
2488     [(156, 8), (158, 8), tenor_five_musicmaker_one],
2489     [(158, 8), (160, 8), tenor_five_musicmaker_one],
2490     [(160, 8), (162, 8), tenor_five_musicmaker_two],
2491     [(162, 8), (164, 8), tenor_five_musicmaker_two],
2492     [(164, 8), (166, 8), tenor_five_musicmaker_one],
2493     [(166, 8), (168, 8), tenor_five_musicmaker_one],
2494     [(168, 8), (170, 8), tenor_five_musicmaker_two],
2495     [(170, 8), (172, 8), tenor_five_musicmaker_one],
2496     [(172, 8), (174, 8), tenor_five_musicmaker_one],
2497     [(174, 8), (176, 8), tenor_five_musicmaker_two],
2498     [(176, 8), (178, 8), tenor_five_musicmaker_two],
2499     [(178, 8), (180, 8), tenor_five_musicmaker_one],
2500     [(180, 8), (182, 8), tenor_five_musicmaker_one],
2501     [(182, 8), (184, 8), tenor_five_musicmaker_one],
2502     [(184, 8), (186, 8), tenor_five_musicmaker_one],
2503     [(186, 8), (188, 8), tenor_five_musicmaker_two],
2504     [(188, 8), (190, 8), tenor_five_musicmaker_two],
2505     [(190, 8), (192, 8), tenor_five_musicmaker_one],
2506     [(192, 8), (194, 8), tenor_five_musicmaker_two],
2507     [(194, 8), (196, 8), tenor_five_musicmaker_one],
2508     [(196, 8), (198, 8), tenor_five_musicmaker_two],
2509     [(198, 8), (199, 8), tenor_five_musicmaker_one],
2510     [(199, 8), (200, 8), tenor_five_musicmaker_one],
2511     ],
2512   ])
2513
2514 voice_16_timespan_list = abjad.TimespanList([
2515   abjad.AnnotatedTimespan(
2516     start_offset=start_offset,
2517     stop_offset=stop_offset,
2518     annotation=MusicSpecifier(
2519       music_maker=music_maker,
2520       voice_name='Voice 16',
2521     ),
2522   )

```

```
2523     for start_offset, stop_offset, music_maker in [
2524         [(0, 8), (2, 8), baritone_one_musicmaker_two],
2525         [(2, 8), (4, 8), baritone_one_musicmaker_two],
2526         [(4, 8), (6, 8), baritone_one_musicmaker_two],
2527         [(6, 8), (8, 8), baritone_one_musicmaker_one],
2528         [(8, 8), (10, 8), baritone_one_musicmaker_one],
2529         [(10, 8), (12, 8), baritone_one_musicmaker_one],
2530         [(12, 8), (14, 8), baritone_one_musicmaker_one],
2531         [(14, 8), (16, 8), baritone_one_musicmaker_one],
2532         [(16, 8), (18, 8), baritone_one_musicmaker_one],
2533         [(18, 8), (20, 8), baritone_one_musicmaker_two],
2534         [(20, 8), (22, 8), baritone_one_musicmaker_two],
2535         [(22, 8), (24, 8), baritone_one_musicmaker_one],
2536         [(24, 8), (26, 8), baritone_one_musicmaker_one],
2537         [(26, 8), (28, 8), baritone_one_musicmaker_one],
2538         [(28, 8), (30, 8), baritone_one_musicmaker_one],
2539         [(30, 8), (32, 8), baritone_one_musicmaker_one],
2540         [(32, 8), (34, 8), baritone_one_musicmaker_two],
2541         [(34, 8), (36, 8), baritone_one_musicmaker_two],
2542         [(36, 8), (38, 8), baritone_one_musicmaker_two],
2543         [(38, 8), (40, 8), baritone_one_musicmaker_one],
2544         [(40, 8), (42, 8), baritone_one_musicmaker_one],
2545         [(42, 8), (44, 8), baritone_one_musicmaker_one],
2546         [(44, 8), (46, 8), baritone_one_musicmaker_one],
2547         [(46, 8), (48, 8), baritone_one_musicmaker_one],
2548         [(48, 8), (50, 8), baritone_one_musicmaker_two],
2549         [(50, 8), (52, 8), baritone_one_musicmaker_one],
2550         [(52, 8), (54, 8), baritone_one_musicmaker_two],
2551         [(54, 8), (56, 8), baritone_one_musicmaker_one],
2552         [(56, 8), (58, 8), baritone_one_musicmaker_one],
2553         [(58, 8), (60, 8), baritone_one_musicmaker_one],
2554         [(60, 8), (62, 8), baritone_one_musicmaker_one],
2555         [(62, 8), (64, 8), baritone_one_musicmaker_one],
2556         [(64, 8), (66, 8), baritone_one_musicmaker_two],
2557         [(66, 8), (68, 8), baritone_one_musicmaker_two],
2558         [(68, 8), (70, 8), baritone_one_musicmaker_one],
2559         [(70, 8), (72, 8), baritone_one_musicmaker_one],
2560         [(72, 8), (74, 8), baritone_one_musicmaker_one],
2561         [(74, 8), (76, 8), baritone_one_musicmaker_one],
2562         [(76, 8), (78, 8), baritone_one_musicmaker_two],
2563         [(78, 8), (80, 8), baritone_one_musicmaker_one],
2564         [(80, 8), (82, 8), baritone_one_musicmaker_one],
2565         [(82, 8), (84, 8), baritone_one_musicmaker_one],
2566         [(84, 8), (86, 8), baritone_one_musicmaker_two],
2567         [(86, 8), (88, 8), baritone_one_musicmaker_one],
2568         [(88, 8), (90, 8), baritone_one_musicmaker_two],
2569         [(90, 8), (92, 8), baritone_one_musicmaker_one],
2570         [(92, 8), (94, 8), baritone_one_musicmaker_two],
2571         [(94, 8), (96, 8), baritone_one_musicmaker_one],
2572         [(96, 8), (98, 8), baritone_one_musicmaker_one],
2573         [(98, 8), (100, 8), baritone_one_musicmaker_one],
2574         [(100, 8), (102, 8), baritone_one_musicmaker_two],
2575         [(102, 8), (104, 8), baritone_one_musicmaker_one],
2576         [(104, 8), (106, 8), baritone_one_musicmaker_two],
2577         [(106, 8), (108, 8), baritone_one_musicmaker_one],
2578         [(108, 8), (110, 8), baritone_one_musicmaker_two],
2579         [(110, 8), (112, 8), baritone_one_musicmaker_one],
2580         [(112, 8), (114, 8), baritone_one_musicmaker_one],
2581         [(114, 8), (116, 8), baritone_one_musicmaker_one],
```

```

2582     [(116, 8), (118, 8), baritone_one_musicmaker_two],
2583     [(118, 8), (120, 8), baritone_one_musicmaker_two],
2584     [(120, 8), (122, 8), baritone_one_musicmaker_two],
2585     [(122, 8), (124, 8), baritone_one_musicmaker_one],
2586     [(124, 8), (126, 8), baritone_one_musicmaker_one],
2587     [(126, 8), (128, 8), baritone_one_musicmaker_one],
2588     [(128, 8), (130, 8), baritone_one_musicmaker_one],
2589     [(130, 8), (132, 8), baritone_one_musicmaker_one],
2590     [(132, 8), (134, 8), baritone_one_musicmaker_one],
2591     [(134, 8), (136, 8), baritone_one_musicmaker_two],
2592     [(136, 8), (138, 8), baritone_one_musicmaker_two],
2593     [(138, 8), (140, 8), baritone_one_musicmaker_two],
2594     [(140, 8), (142, 8), baritone_one_musicmaker_two],
2595     [(142, 8), (144, 8), baritone_one_musicmaker_one],
2596     [(144, 8), (146, 8), baritone_one_musicmaker_one],
2597     [(146, 8), (148, 8), baritone_one_musicmaker_one],
2598     [(148, 8), (150, 8), baritone_one_musicmaker_one],
2599     [(150, 8), (152, 8), baritone_one_musicmaker_two],
2600     [(152, 8), (154, 8), baritone_one_musicmaker_two],
2601     [(154, 8), (156, 8), baritone_one_musicmaker_two],
2602     [(156, 8), (158, 8), baritone_one_musicmaker_one],
2603     [(158, 8), (160, 8), baritone_one_musicmaker_one],
2604     [(160, 8), (162, 8), baritone_one_musicmaker_one],
2605     [(162, 8), (164, 8), baritone_one_musicmaker_two],
2606     [(164, 8), (166, 8), baritone_one_musicmaker_one],
2607     [(166, 8), (168, 8), baritone_one_musicmaker_one],
2608     [(168, 8), (170, 8), baritone_one_musicmaker_two],
2609     [(170, 8), (172, 8), baritone_one_musicmaker_one],
2610     [(172, 8), (174, 8), baritone_one_musicmaker_one],
2611     [(174, 8), (176, 8), baritone_one_musicmaker_one],
2612     [(176, 8), (178, 8), baritone_one_musicmaker_two],
2613     [(178, 8), (180, 8), baritone_one_musicmaker_two],
2614     [(180, 8), (182, 8), baritone_one_musicmaker_one],
2615     [(182, 8), (184, 8), baritone_one_musicmaker_one],
2616     [(184, 8), (186, 8), baritone_one_musicmaker_one],
2617     [(186, 8), (188, 8), baritone_one_musicmaker_one],
2618     [(188, 8), (190, 8), baritone_one_musicmaker_one],
2619     [(190, 8), (192, 8), baritone_one_musicmaker_two],
2620     [(192, 8), (194, 8), baritone_one_musicmaker_two],
2621     [(194, 8), (196, 8), baritone_one_musicmaker_one],
2622     [(196, 8), (198, 8), baritone_one_musicmaker_one],
2623     [(198, 8), (199, 8), baritone_one_musicmaker_two],
2624     [(199, 8), (200, 8), baritone_one_musicmaker_one],
2625   ],
2626 ])
2627
2628 voice_17_timespan_list = abjad.TimespanList([
2629   abjad.AnnotatedTimespan(
2630     start_offset=start_offset,
2631     stop_offset=stop_offset,
2632     annotation=MusicSpecifier(
2633       music_maker=music_maker,
2634       voice_name='Voice 17',
2635     ),
2636   )
2637   for start_offset, stop_offset, music_maker in [
2638     [(0, 8), (2, 8), baritone_two_musicmaker_one],
2639     [(2, 8), (4, 8), baritone_two_musicmaker_one],
2640     [(4, 8), (6, 8), baritone_two_musicmaker_two],

```

```
2641 [(6, 8), (8, 8), baritone_two_musicmaker_one],  
2642 [(8, 8), (10, 8), baritone_two_musicmaker_one],  
2643 [(10, 8), (12, 8), baritone_two_musicmaker_two],  
2644 [(12, 8), (14, 8), baritone_two_musicmaker_one],  
2645 [(14, 8), (16, 8), baritone_two_musicmaker_one],  
2646 [(16, 8), (18, 8), baritone_two_musicmaker_one],  
2647 [(18, 8), (20, 8), baritone_two_musicmaker_two],  
2648 [(20, 8), (22, 8), baritone_two_musicmaker_one],  
2649 [(22, 8), (24, 8), baritone_two_musicmaker_one],  
2650 [(24, 8), (26, 8), baritone_two_musicmaker_two],  
2651 [(26, 8), (28, 8), baritone_two_musicmaker_two],  
2652 [(28, 8), (30, 8), baritone_two_musicmaker_one],  
2653 [(30, 8), (32, 8), baritone_two_musicmaker_one],  
2654 [(32, 8), (34, 8), baritone_two_musicmaker_one],  
2655 [(34, 8), (36, 8), baritone_two_musicmaker_two],  
2656 [(36, 8), (38, 8), baritone_two_musicmaker_two],  
2657 [(38, 8), (40, 8), baritone_two_musicmaker_two],  
2658 [(40, 8), (42, 8), baritone_two_musicmaker_one],  
2659 [(42, 8), (44, 8), baritone_two_musicmaker_two],  
2660 [(44, 8), (46, 8), baritone_two_musicmaker_two],  
2661 [(46, 8), (48, 8), baritone_two_musicmaker_one],  
2662 [(48, 8), (50, 8), baritone_two_musicmaker_one],  
2663 [(50, 8), (52, 8), baritone_two_musicmaker_two],  
2664 [(52, 8), (54, 8), baritone_two_musicmaker_two],  
2665 [(54, 8), (56, 8), baritone_two_musicmaker_one],  
2666 [(56, 8), (58, 8), baritone_two_musicmaker_one],  
2667 [(58, 8), (60, 8), baritone_two_musicmaker_one],  
2668 [(60, 8), (62, 8), baritone_two_musicmaker_two],  
2669 [(62, 8), (64, 8), baritone_two_musicmaker_two],  
2670 [(64, 8), (66, 8), baritone_two_musicmaker_two],  
2671 [(66, 8), (68, 8), baritone_two_musicmaker_one],  
2672 [(68, 8), (70, 8), baritone_two_musicmaker_two],  
2673 [(70, 8), (72, 8), baritone_two_musicmaker_two],  
2674 [(72, 8), (74, 8), baritone_two_musicmaker_one],  
2675 [(74, 8), (76, 8), baritone_two_musicmaker_one],  
2676 [(76, 8), (78, 8), baritone_two_musicmaker_one],  
2677 [(78, 8), (80, 8), baritone_two_musicmaker_two],  
2678 [(80, 8), (82, 8), baritone_two_musicmaker_one],  
2679 [(82, 8), (84, 8), baritone_two_musicmaker_two],  
2680 [(84, 8), (86, 8), baritone_two_musicmaker_one],  
2681 [(86, 8), (88, 8), baritone_two_musicmaker_two],  
2682 [(88, 8), (90, 8), baritone_two_musicmaker_two],  
2683 [(90, 8), (92, 8), baritone_two_musicmaker_one],  
2684 [(92, 8), (94, 8), baritone_two_musicmaker_one],  
2685 [(94, 8), (96, 8), baritone_two_musicmaker_one],  
2686 [(96, 8), (98, 8), baritone_two_musicmaker_two],  
2687 [(98, 8), (100, 8), baritone_two_musicmaker_two],  
2688 [(100, 8), (102, 8), baritone_two_musicmaker_two],  
2689 [(102, 8), (104, 8), baritone_two_musicmaker_one],  
2690 [(104, 8), (106, 8), baritone_two_musicmaker_two],  
2691 [(106, 8), (108, 8), baritone_two_musicmaker_one],  
2692 [(108, 8), (110, 8), baritone_two_musicmaker_two],  
2693 [(110, 8), (112, 8), baritone_two_musicmaker_two],  
2694 [(112, 8), (114, 8), baritone_two_musicmaker_one],  
2695 [(114, 8), (116, 8), baritone_two_musicmaker_one],  
2696 [(116, 8), (118, 8), baritone_two_musicmaker_one],  
2697 [(118, 8), (120, 8), baritone_two_musicmaker_two],  
2698 [(120, 8), (122, 8), baritone_two_musicmaker_two],  
2699 [(122, 8), (124, 8), baritone_two_musicmaker_two],
```

```

2700     [(124, 8), (126, 8), baritone_two_musicmaker_one],
2701     [(126, 8), (128, 8), baritone_two_musicmaker_one],
2702     [(128, 8), (130, 8), baritone_two_musicmaker_one],
2703     [(130, 8), (132, 8), baritone_two_musicmaker_two],
2704     [(132, 8), (134, 8), baritone_two_musicmaker_two],
2705     [(134, 8), (136, 8), baritone_two_musicmaker_two],
2706     [(136, 8), (138, 8), baritone_two_musicmaker_one],
2707     [(138, 8), (140, 8), baritone_two_musicmaker_one],
2708     [(140, 8), (142, 8), baritone_two_musicmaker_one],
2709     [(142, 8), (144, 8), baritone_two_musicmaker_one],
2710     [(144, 8), (146, 8), baritone_two_musicmaker_two],
2711     [(146, 8), (148, 8), baritone_two_musicmaker_one],
2712     [(148, 8), (150, 8), baritone_two_musicmaker_one],
2713     [(150, 8), (152, 8), baritone_two_musicmaker_two],
2714     [(152, 8), (154, 8), baritone_two_musicmaker_one],
2715     [(154, 8), (156, 8), baritone_two_musicmaker_one],
2716     [(156, 8), (158, 8), baritone_two_musicmaker_two],
2717     [(158, 8), (160, 8), baritone_two_musicmaker_two],
2718     [(160, 8), (162, 8), baritone_two_musicmaker_one],
2719     [(162, 8), (164, 8), baritone_two_musicmaker_two],
2720     [(164, 8), (166, 8), baritone_two_musicmaker_one],
2721     [(166, 8), (168, 8), baritone_two_musicmaker_one],
2722     [(168, 8), (170, 8), baritone_two_musicmaker_two],
2723     [(170, 8), (172, 8), baritone_two_musicmaker_two],
2724     [(172, 8), (174, 8), baritone_two_musicmaker_one],
2725     [(174, 8), (176, 8), baritone_two_musicmaker_one],
2726     [(176, 8), (178, 8), baritone_two_musicmaker_one],
2727     [(178, 8), (180, 8), baritone_two_musicmaker_two],
2728     [(180, 8), (182, 8), baritone_two_musicmaker_one],
2729     [(182, 8), (184, 8), baritone_two_musicmaker_one],
2730     [(184, 8), (186, 8), baritone_two_musicmaker_one],
2731     [(186, 8), (188, 8), baritone_two_musicmaker_two],
2732     [(188, 8), (190, 8), baritone_two_musicmaker_one],
2733     [(190, 8), (192, 8), baritone_two_musicmaker_two],
2734     [(192, 8), (194, 8), baritone_two_musicmaker_one],
2735     [(194, 8), (196, 8), baritone_two_musicmaker_two],
2736     [(196, 8), (198, 8), baritone_two_musicmaker_one],
2737     [(198, 8), (199, 8), baritone_two_musicmaker_two],
2738     [(199, 8), (200, 8), baritone_two_musicmaker_one],
2739   ]
2740 ])
2741
2742 voice_18_timespan_list = abjad.TimespanList([
2743     abjad.AnnotatedTimespan(
2744         start_offset=start_offset,
2745         stop_offset=stop_offset,
2746         annotation=MusicSpecifier(
2747             music_maker=music_maker,
2748             voice_name='Voice 18',
2749         ),
2750     )
2751     for start_offset, stop_offset, music_maker in [
2752         [(0, 8), (2, 8), baritone_three_musicmaker_one],
2753         [(2, 8), (4, 8), baritone_three_musicmaker_one],
2754         [(4, 8), (6, 8), baritone_three_musicmaker_one],
2755         [(6, 8), (8, 8), baritone_three_musicmaker_two],
2756         [(8, 8), (10, 8), baritone_three_musicmaker_two],
2757         [(10, 8), (12, 8), baritone_three_musicmaker_one],
2758         [(12, 8), (14, 8), baritone_three_musicmaker_one],

```

```
2759 [(14, 8), (16, 8), baritone_three_musicmaker_one],  
2760 [(16, 8), (18, 8), baritone_three_musicmaker_one],  
2761 [(18, 8), (20, 8), baritone_three_musicmaker_one],  
2762 [(20, 8), (22, 8), baritone_three_musicmaker_two],  
2763 [(22, 8), (24, 8), baritone_three_musicmaker_two],  
2764 [(24, 8), (26, 8), baritone_three_musicmaker_one],  
2765 [(26, 8), (28, 8), baritone_three_musicmaker_one],  
2766 [(28, 8), (30, 8), baritone_three_musicmaker_one],  
2767 [(30, 8), (32, 8), baritone_three_musicmaker_two],  
2768 [(32, 8), (34, 8), baritone_three_musicmaker_two],  
2769 [(34, 8), (36, 8), baritone_three_musicmaker_two],  
2770 [(36, 8), (38, 8), baritone_three_musicmaker_one],  
2771 [(38, 8), (40, 8), baritone_three_musicmaker_one],  
2772 [(40, 8), (42, 8), baritone_three_musicmaker_one],  
2773 [(42, 8), (44, 8), baritone_three_musicmaker_two],  
2774 [(44, 8), (46, 8), baritone_three_musicmaker_one],  
2775 [(46, 8), (48, 8), baritone_three_musicmaker_two],  
2776 [(48, 8), (50, 8), baritone_three_musicmaker_two],  
2777 [(50, 8), (52, 8), baritone_three_musicmaker_one],  
2778 [(52, 8), (54, 8), baritone_three_musicmaker_one],  
2779 [(54, 8), (56, 8), baritone_three_musicmaker_one],  
2780 [(56, 8), (58, 8), baritone_three_musicmaker_two],  
2781 [(58, 8), (60, 8), baritone_three_musicmaker_one],  
2782 [(60, 8), (62, 8), baritone_three_musicmaker_two],  
2783 [(62, 8), (64, 8), baritone_three_musicmaker_two],  
2784 [(64, 8), (66, 8), baritone_three_musicmaker_two],  
2785 [(66, 8), (68, 8), baritone_three_musicmaker_one],  
2786 [(68, 8), (70, 8), baritone_three_musicmaker_one],  
2787 [(70, 8), (72, 8), baritone_three_musicmaker_one],  
2788 [(72, 8), (74, 8), baritone_three_musicmaker_two],  
2789 [(74, 8), (76, 8), baritone_three_musicmaker_two],  
2790 [(76, 8), (78, 8), baritone_three_musicmaker_one],  
2791 [(78, 8), (80, 8), baritone_three_musicmaker_one],  
2792 [(80, 8), (82, 8), baritone_three_musicmaker_two],  
2793 [(82, 8), (84, 8), baritone_three_musicmaker_two],  
2794 [(84, 8), (86, 8), baritone_three_musicmaker_one],  
2795 [(86, 8), (88, 8), baritone_three_musicmaker_one],  
2796 [(88, 8), (90, 8), baritone_three_musicmaker_two],  
2797 [(90, 8), (92, 8), baritone_three_musicmaker_two],  
2798 [(92, 8), (94, 8), baritone_three_musicmaker_one],  
2799 [(94, 8), (96, 8), baritone_three_musicmaker_one],  
2800 [(96, 8), (98, 8), baritone_three_musicmaker_one],  
2801 [(98, 8), (100, 8), baritone_three_musicmaker_one],  
2802 [(100, 8), (102, 8), baritone_three_musicmaker_two],  
2803 [(102, 8), (104, 8), baritone_three_musicmaker_two],  
2804 [(104, 8), (106, 8), baritone_three_musicmaker_one],  
2805 [(106, 8), (108, 8), baritone_three_musicmaker_one],  
2806 [(108, 8), (110, 8), baritone_three_musicmaker_one],  
2807 [(110, 8), (112, 8), baritone_three_musicmaker_one],  
2808 [(112, 8), (114, 8), baritone_three_musicmaker_one],  
2809 [(114, 8), (116, 8), baritone_three_musicmaker_one],  
2810 [(116, 8), (118, 8), baritone_three_musicmaker_two],  
2811 [(118, 8), (120, 8), baritone_three_musicmaker_two],  
2812 [(120, 8), (122, 8), baritone_three_musicmaker_one],  
2813 [(122, 8), (124, 8), baritone_three_musicmaker_one],  
2814 [(124, 8), (126, 8), baritone_three_musicmaker_one],  
2815 [(126, 8), (128, 8), baritone_three_musicmaker_two],  
2816 [(128, 8), (130, 8), baritone_three_musicmaker_two],  
2817 [(130, 8), (132, 8), baritone_three_musicmaker_one],
```

```

2818     [(132, 8), (134, 8), baritone_three_musicmaker_one],
2819     [(134, 8), (136, 8), baritone_three_musicmaker_two],
2820     [(136, 8), (138, 8), baritone_three_musicmaker_one],
2821     [(138, 8), (140, 8), baritone_three_musicmaker_one],
2822     [(140, 8), (142, 8), baritone_three_musicmaker_two],
2823     [(142, 8), (144, 8), baritone_three_musicmaker_one],
2824     [(144, 8), (146, 8), baritone_three_musicmaker_one],
2825     [(146, 8), (148, 8), baritone_three_musicmaker_two],
2826     [(148, 8), (150, 8), baritone_three_musicmaker_two],
2827     [(150, 8), (152, 8), baritone_three_musicmaker_one],
2828     [(152, 8), (154, 8), baritone_three_musicmaker_two],
2829     [(154, 8), (156, 8), baritone_three_musicmaker_two],
2830     [(156, 8), (158, 8), baritone_three_musicmaker_one],
2831     [(158, 8), (160, 8), baritone_three_musicmaker_one],
2832     [(160, 8), (162, 8), baritone_three_musicmaker_two],
2833     [(162, 8), (164, 8), baritone_three_musicmaker_one],
2834     [(164, 8), (166, 8), baritone_three_musicmaker_two],
2835     [(166, 8), (168, 8), baritone_three_musicmaker_one],
2836     [(168, 8), (170, 8), baritone_three_musicmaker_two],
2837     [(170, 8), (172, 8), baritone_three_musicmaker_two],
2838     [(172, 8), (174, 8), baritone_three_musicmaker_one],
2839     [(174, 8), (176, 8), baritone_three_musicmaker_two],
2840     [(176, 8), (178, 8), baritone_three_musicmaker_one],
2841     [(178, 8), (180, 8), baritone_three_musicmaker_one],
2842     [(180, 8), (182, 8), baritone_three_musicmaker_two],
2843     [(182, 8), (184, 8), baritone_three_musicmaker_two],
2844     [(184, 8), (186, 8), baritone_three_musicmaker_one],
2845     [(186, 8), (188, 8), baritone_three_musicmaker_one],
2846     [(188, 8), (190, 8), baritone_three_musicmaker_one],
2847     [(190, 8), (192, 8), baritone_three_musicmaker_two],
2848     [(192, 8), (194, 8), baritone_three_musicmaker_two],
2849     [(194, 8), (196, 8), baritone_three_musicmaker_one],
2850     [(196, 8), (198, 8), baritone_three_musicmaker_one],
2851     [(198, 8), (199, 8), baritone_three_musicmaker_two],
2852     [(199, 8), (200, 8), baritone_three_musicmaker_one],
2853 ]
2854 ])
2855
2856 voice_19_timespan_list = abjad.TimespanList([
2857     abjad.AnnotatedTimespan(
2858         start_offset=start_offset,
2859         stop_offset=stop_offset,
2860         annotation=MusicSpecifier(
2861             music_maker=music_maker,
2862             voice_name='Voice 19',
2863         ),
2864     )
2865     for start_offset, stop_offset, music_maker in [
2866         [(0, 8), (2, 8), bass_one_musicmaker_one],
2867         [(2, 8), (4, 8), bass_one_musicmaker_two],
2868         [(4, 8), (6, 8), bass_one_musicmaker_one],
2869         [(6, 8), (8, 8), bass_one_musicmaker_two],
2870         [(8, 8), (10, 8), bass_one_musicmaker_one],
2871         [(10, 8), (12, 8), bass_one_musicmaker_two],
2872         [(12, 8), (14, 8), bass_one_musicmaker_one],
2873         [(14, 8), (16, 8), bass_one_musicmaker_one],
2874         [(16, 8), (18, 8), bass_one_musicmaker_two],
2875         [(18, 8), (20, 8), bass_one_musicmaker_two],
2876         [(20, 8), (22, 8), bass_one_musicmaker_one],

```

```
2877 [(22, 8), (24, 8), bass_one_musicmaker_two],  
2878 [(24, 8), (26, 8), bass_one_musicmaker_two],  
2879 [(26, 8), (28, 8), bass_one_musicmaker_two],  
2880 [(28, 8), (30, 8), bass_one_musicmaker_one],  
2881 [(30, 8), (32, 8), bass_one_musicmaker_one],  
2882 [(32, 8), (34, 8), bass_one_musicmaker_one],  
2883 [(34, 8), (36, 8), bass_one_musicmaker_two],  
2884 [(36, 8), (38, 8), bass_one_musicmaker_two],  
2885 [(38, 8), (40, 8), bass_one_musicmaker_one],  
2886 [(40, 8), (42, 8), bass_one_musicmaker_one],  
2887 [(42, 8), (44, 8), bass_one_musicmaker_two],  
2888 [(44, 8), (46, 8), bass_one_musicmaker_two],  
2889 [(46, 8), (48, 8), bass_one_musicmaker_two],  
2890 [(48, 8), (50, 8), bass_one_musicmaker_one],  
2891 [(50, 8), (52, 8), bass_one_musicmaker_one],  
2892 [(52, 8), (54, 8), bass_one_musicmaker_one],  
2893 [(54, 8), (56, 8), bass_one_musicmaker_two],  
2894 [(56, 8), (58, 8), bass_one_musicmaker_two],  
2895 [(58, 8), (60, 8), bass_one_musicmaker_one],  
2896 [(60, 8), (62, 8), bass_one_musicmaker_one],  
2897 [(62, 8), (64, 8), bass_one_musicmaker_two],  
2898 [(64, 8), (68, 8), bass_one_musicmaker_two],  
2899 [(68, 8), (70, 8), bass_one_musicmaker_one],  
2900 [(70, 8), (72, 8), bass_one_musicmaker_one],  
2901 [(72, 8), (74, 8), bass_one_musicmaker_two],  
2902 [(74, 8), (76, 8), bass_one_musicmaker_one],  
2903 [(76, 8), (78, 8), bass_one_musicmaker_one],  
2904 [(78, 8), (80, 8), bass_one_musicmaker_one],  
2905 [(80, 8), (82, 8), bass_one_musicmaker_two],  
2906 [(82, 8), (84, 8), bass_one_musicmaker_one],  
2907 [(84, 8), (86, 8), bass_one_musicmaker_one],  
2908 [(86, 8), (88, 8), bass_one_musicmaker_one],  
2909 [(88, 8), (90, 8), bass_one_musicmaker_two],  
2910 [(90, 8), (92, 8), bass_one_musicmaker_one],  
2911 [(92, 8), (94, 8), bass_one_musicmaker_one],  
2912 [(94, 8), (96, 8), bass_one_musicmaker_one],  
2913 [(96, 8), (98, 8), bass_one_musicmaker_one],  
2914 [(98, 8), (100, 8), bass_one_musicmaker_two],  
2915 [(100, 8), (102, 8), bass_one_musicmaker_one],  
2916 [(102, 8), (104, 8), bass_one_musicmaker_one],  
2917 [(104, 8), (106, 8), bass_one_musicmaker_one],  
2918 [(106, 8), (108, 8), bass_one_musicmaker_two],  
2919 [(108, 8), (110, 8), bass_one_musicmaker_two],  
2920 [(110, 8), (112, 8), bass_one_musicmaker_two],  
2921 [(112, 8), (114, 8), bass_one_musicmaker_one],  
2922 [(114, 8), (116, 8), bass_one_musicmaker_one],  
2923 [(116, 8), (118, 8), bass_one_musicmaker_one],  
2924 [(118, 8), (120, 8), bass_one_musicmaker_two],  
2925 [(120, 8), (122, 8), bass_one_musicmaker_one],  
2926 [(122, 8), (124, 8), bass_one_musicmaker_one],  
2927 [(124, 8), (126, 8), bass_one_musicmaker_two],  
2928 [(126, 8), (128, 8), bass_one_musicmaker_one],  
2929 [(128, 8), (130, 8), bass_one_musicmaker_one],  
2930 [(130, 8), (132, 8), bass_one_musicmaker_one],  
2931 [(132, 8), (134, 8), bass_one_musicmaker_one],  
2932 [(134, 8), (136, 8), bass_one_musicmaker_two],  
2933 [(136, 8), (138, 8), bass_one_musicmaker_one],  
2934 [(138, 8), (140, 8), bass_one_musicmaker_one],  
2935 [(140, 8), (142, 8), bass_one_musicmaker_one],
```

```

2936     [(142, 8), (144, 8), bass_one_musicmaker_two],
2937     [(144, 8), (146, 8), bass_one_musicmaker_two],
2938     [(146, 8), (148, 8), bass_one_musicmaker_two],
2939     [(148, 8), (150, 8), bass_one_musicmaker_one],
2940     [(150, 8), (152, 8), bass_one_musicmaker_one],
2941     [(152, 8), (154, 8), bass_one_musicmaker_one],
2942     [(154, 8), (156, 8), bass_one_musicmaker_two],
2943     [(156, 8), (158, 8), bass_one_musicmaker_one],
2944     [(158, 8), (160, 8), bass_one_musicmaker_one],
2945     [(160, 8), (162, 8), bass_one_musicmaker_two],
2946     [(162, 8), (164, 8), bass_one_musicmaker_one],
2947     [(164, 8), (168, 8), bass_one_musicmaker_one],
2948     [(168, 8), (170, 8), bass_one_musicmaker_one],
2949     [(170, 8), (172, 8), bass_one_musicmaker_one],
2950     [(172, 8), (174, 8), bass_one_musicmaker_one],
2951     [(174, 8), (176, 8), bass_one_musicmaker_two],
2952     [(176, 8), (178, 8), bass_one_musicmaker_two],
2953     [(178, 8), (180, 8), bass_one_musicmaker_one],
2954     [(180, 8), (182, 8), bass_one_musicmaker_one],
2955     [(182, 8), (184, 8), bass_one_musicmaker_one],
2956     [(184, 8), (186, 8), bass_one_musicmaker_one],
2957     [(186, 8), (188, 8), bass_one_musicmaker_one],
2958     [(188, 8), (190, 8), bass_one_musicmaker_one],
2959     [(190, 8), (192, 8), bass_one_musicmaker_two],
2960     [(192, 8), (194, 8), bass_one_musicmaker_two],
2961     [(194, 8), (196, 8), bass_one_musicmaker_two],
2962     [(196, 8), (198, 8), bass_one_musicmaker_one],
2963     [(198, 8), (199, 8), bass_one_musicmaker_one],
2964     [(199, 8), (200, 8), bass_one_musicmaker_two],
2965   ]
2966 ])
2967
2968 voice_20_timestrap_list = abjad.TimespanList([
2969   abjad.AnnotatedTimespan(
2970     start_offset=start_offset,
2971     stop_offset=stop_offset,
2972     annotation=MusicSpecifier(
2973       music_maker=music_maker,
2974       voice_name='Voice 20',
2975     ),
2976   )
2977   for start_offset, stop_offset, music_maker in [
2978     [(0, 8), (2, 8), bass_two_musicmaker_one],
2979     [(2, 8), (4, 8), bass_two_musicmaker_one],
2980     [(4, 8), (6, 8), bass_two_musicmaker_two],
2981     [(6, 8), (8, 8), bass_two_musicmaker_one],
2982     [(8, 8), (10, 8), bass_two_musicmaker_two],
2983     [(10, 8), (12, 8), bass_two_musicmaker_two],
2984     [(12, 8), (14, 8), bass_two_musicmaker_one],
2985     [(14, 8), (16, 8), bass_two_musicmaker_two],
2986     [(16, 8), (18, 8), bass_two_musicmaker_one],
2987     [(18, 8), (20, 8), bass_two_musicmaker_one],
2988     [(20, 8), (22, 8), bass_two_musicmaker_one],
2989     [(22, 8), (24, 8), bass_two_musicmaker_two],
2990     [(24, 8), (26, 8), bass_two_musicmaker_two],
2991     [(26, 8), (28, 8), bass_two_musicmaker_one],
2992     [(28, 8), (30, 8), bass_two_musicmaker_one],
2993     [(30, 8), (32, 8), bass_two_musicmaker_two],
2994     [(32, 8), (34, 8), bass_two_musicmaker_one],

```

```
2995 [(34, 8), (36, 8), bass_two_musicmaker_one],  
2996 [(36, 8), (38, 8), bass_two_musicmaker_one],  
2997 [(38, 8), (40, 8), bass_two_musicmaker_two],  
2998 [(40, 8), (42, 8), bass_two_musicmaker_one],  
2999 [(42, 8), (44, 8), bass_two_musicmaker_one],  
3000 [(44, 8), (46, 8), bass_two_musicmaker_one],  
3001 [(46, 8), (48, 8), bass_two_musicmaker_one],  
3002 [(48, 8), (50, 8), bass_two_musicmaker_one],  
3003 [(50, 8), (52, 8), bass_two_musicmaker_two],  
3004 [(52, 8), (54, 8), bass_two_musicmaker_two],  
3005 [(54, 8), (56, 8), bass_two_musicmaker_one],  
3006 [(56, 8), (58, 8), bass_two_musicmaker_two],  
3007 [(58, 8), (60, 8), bass_two_musicmaker_one],  
3008 [(60, 8), (62, 8), bass_two_musicmaker_one],  
3009 [(62, 8), (64, 8), bass_two_musicmaker_one],  
3010 [(64, 8), (68, 8), bass_two_musicmaker_one],  
3011 [(68, 8), (70, 8), bass_two_musicmaker_one],  
3012 [(70, 8), (72, 8), bass_two_musicmaker_one],  
3013 [(72, 8), (74, 8), bass_two_musicmaker_one],  
3014 [(74, 8), (76, 8), bass_two_musicmaker_two],  
3015 [(76, 8), (78, 8), bass_two_musicmaker_two],  
3016 [(78, 8), (80, 8), bass_two_musicmaker_two],  
3017 [(80, 8), (82, 8), bass_two_musicmaker_one],  
3018 [(82, 8), (84, 8), bass_two_musicmaker_one],  
3019 [(84, 8), (86, 8), bass_two_musicmaker_one],  
3020 [(86, 8), (88, 8), bass_two_musicmaker_one],  
3021 [(88, 8), (90, 8), bass_two_musicmaker_one],  
3022 [(90, 8), (92, 8), bass_two_musicmaker_two],  
3023 [(92, 8), (94, 8), bass_two_musicmaker_two],  
3024 [(94, 8), (96, 8), bass_two_musicmaker_one],  
3025 [(96, 8), (98, 8), bass_two_musicmaker_one],  
3026 [(98, 8), (100, 8), bass_two_musicmaker_two],  
3027 [(100, 8), (102, 8), bass_two_musicmaker_one],  
3028 [(102, 8), (104, 8), bass_two_musicmaker_one],  
3029 [(104, 8), (106, 8), bass_two_musicmaker_one],  
3030 [(106, 8), (108, 8), bass_two_musicmaker_two],  
3031 [(108, 8), (110, 8), bass_two_musicmaker_two],  
3032 [(110, 8), (112, 8), bass_two_musicmaker_one],  
3033 [(112, 8), (114, 8), bass_two_musicmaker_one],  
3034 [(114, 8), (116, 8), bass_two_musicmaker_one],  
3035 [(116, 8), (118, 8), bass_two_musicmaker_one],  
3036 [(118, 8), (120, 8), bass_two_musicmaker_two],  
3037 [(120, 8), (122, 8), bass_two_musicmaker_one],  
3038 [(122, 8), (124, 8), bass_two_musicmaker_one],  
3039 [(124, 8), (126, 8), bass_two_musicmaker_two],  
3040 [(126, 8), (128, 8), bass_two_musicmaker_one],  
3041 [(128, 8), (130, 8), bass_two_musicmaker_one],  
3042 [(130, 8), (132, 8), bass_two_musicmaker_one],  
3043 [(132, 8), (134, 8), bass_two_musicmaker_one],  
3044 [(134, 8), (136, 8), bass_two_musicmaker_one],  
3045 [(136, 8), (138, 8), bass_two_musicmaker_two],  
3046 [(138, 8), (140, 8), bass_two_musicmaker_one],  
3047 [(140, 8), (142, 8), bass_two_musicmaker_one],  
3048 [(142, 8), (144, 8), bass_two_musicmaker_two],  
3049 [(144, 8), (146, 8), bass_two_musicmaker_one],  
3050 [(146, 8), (148, 8), bass_two_musicmaker_one],  
3051 [(148, 8), (150, 8), bass_two_musicmaker_one],  
3052 [(150, 8), (152, 8), bass_two_musicmaker_one],  
3053 [(152, 8), (154, 8), bass_two_musicmaker_two],
```

```

3054     [(154, 8), (156, 8), bass_two_musicmaker_one],
3055     [(156, 8), (158, 8), bass_two_musicmaker_one],
3056     [(158, 8), (160, 8), bass_two_musicmaker_one],
3057     [(160, 8), (162, 8), bass_two_musicmaker_one],
3058     [(162, 8), (164, 8), bass_two_musicmaker_one],
3059     [(164, 8), (168, 8), bass_two_musicmaker_one],
3060     [(168, 8), (170, 8), bass_two_musicmaker_one],
3061     [(170, 8), (172, 8), bass_two_musicmaker_two],
3062     [(172, 8), (174, 8), bass_two_musicmaker_two],
3063     [(174, 8), (176, 8), bass_two_musicmaker_two],
3064     [(176, 8), (178, 8), bass_two_musicmaker_one],
3065     [(178, 8), (180, 8), bass_two_musicmaker_one],
3066     [(180, 8), (182, 8), bass_two_musicmaker_one],
3067     [(182, 8), (184, 8), bass_two_musicmaker_one],
3068     [(184, 8), (186, 8), bass_two_musicmaker_two],
3069     [(186, 8), (188, 8), bass_two_musicmaker_one],
3070     [(188, 8), (190, 8), bass_two_musicmaker_two],
3071     [(190, 8), (192, 8), bass_two_musicmaker_one],
3072     [(192, 8), (194, 8), bass_two_musicmaker_one],
3073     [(194, 8), (196, 8), bass_two_musicmaker_one],
3074     [(196, 8), (198, 8), bass_two_musicmaker_one],
3075     [(198, 8), (199, 8), bass_two_musicmaker_two],
3076     [(199, 8), (200, 8), bass_two_musicmaker_one],
3077 ]
3078 ])
3079
3080 voice_21_timestspan_list = abjad.TimespanList([
3081     abjad.AnnotatedTimespan(
3082         start_offset=start_offset,
3083         stop_offset=stop_offset,
3084         annotation=MusicSpecifier(
3085             music_maker=music_maker,
3086             voice_name='Voice 21',
3087         ),
3088     ),
3089     for start_offset, stop_offset, music_maker in [
3090         [(0, 8), (2, 8), contrabass_musicmaker_one],
3091         [(2, 8), (4, 8), contrabass_musicmaker_one],
3092         [(4, 8), (6, 8), contrabass_musicmaker_one],
3093         [(6, 8), (8, 8), contrabass_musicmaker_two],
3094         [(8, 8), (10, 8), contrabass_musicmaker_two],
3095         [(10, 8), (12, 8), contrabass_musicmaker_one],
3096         [(12, 8), (14, 8), contrabass_musicmaker_one],
3097         [(14, 8), (16, 8), contrabass_musicmaker_one],
3098         [(16, 8), (18, 8), contrabass_musicmaker_one],
3099         [(18, 8), (20, 8), contrabass_musicmaker_one],
3100         [(20, 8), (22, 8), contrabass_musicmaker_two],
3101         [(22, 8), (24, 8), contrabass_musicmaker_two],
3102         [(24, 8), (26, 8), contrabass_musicmaker_one],
3103         [(26, 8), (28, 8), contrabass_musicmaker_one],
3104         [(28, 8), (30, 8), contrabass_musicmaker_one],
3105         [(30, 8), (32, 8), contrabass_musicmaker_one],
3106         [(32, 8), (34, 8), contrabass_musicmaker_one],
3107         [(34, 8), (36, 8), contrabass_musicmaker_one],
3108         [(36, 8), (38, 8), contrabass_musicmaker_one],
3109         [(38, 8), (40, 8), contrabass_musicmaker_two],
3110         [(40, 8), (42, 8), contrabass_musicmaker_two],
3111         [(42, 8), (44, 8), contrabass_musicmaker_one],
3112         [(44, 8), (46, 8), contrabass_musicmaker_one],

```

```
3113 [(46, 8), (48, 8), contrabass_musicmaker_one],  
3114 [(48, 8), (50, 8), contrabass_musicmaker_one],  
3115 [(50, 8), (52, 8), contrabass_musicmaker_one],  
3116 [(52, 8), (54, 8), contrabass_musicmaker_two],  
3117 [(54, 8), (56, 8), contrabass_musicmaker_two],  
3118 [(56, 8), (58, 8), contrabass_musicmaker_two],  
3119 [(58, 8), (60, 8), contrabass_musicmaker_one],  
3120 [(60, 8), (62, 8), contrabass_musicmaker_one],  
3121 [(62, 8), (64, 8), contrabass_musicmaker_one],  
3122 [(64, 8), (68, 8), contrabass_musicmaker_one],  
3123 [(68, 8), (70, 8), contrabass_musicmaker_one],  
3124 [(70, 8), (72, 8), contrabass_musicmaker_two],  
3125 [(72, 8), (74, 8), contrabass_musicmaker_two],  
3126 [(74, 8), (76, 8), contrabass_musicmaker_one],  
3127 [(76, 8), (78, 8), contrabass_musicmaker_one],  
3128 [(78, 8), (80, 8), contrabass_musicmaker_one],  
3129 [(80, 8), (82, 8), contrabass_musicmaker_one],  
3130 [(82, 8), (84, 8), contrabass_musicmaker_one],  
3131 [(84, 8), (86, 8), contrabass_musicmaker_one],  
3132 [(86, 8), (88, 8), contrabass_musicmaker_two],  
3133 [(88, 8), (90, 8), contrabass_musicmaker_two],  
3134 [(90, 8), (92, 8), contrabass_musicmaker_one],  
3135 [(92, 8), (94, 8), contrabass_musicmaker_one],  
3136 [(94, 8), (96, 8), contrabass_musicmaker_one],  
3137 [(96, 8), (98, 8), contrabass_musicmaker_one],  
3138 [(98, 8), (100, 8), contrabass_musicmaker_one],  
3139 [(100, 8), (102, 8), contrabass_musicmaker_two],  
3140 [(102, 8), (104, 8), contrabass_musicmaker_two],  
3141 [(104, 8), (106, 8), contrabass_musicmaker_one],  
3142 [(106, 8), (108, 8), contrabass_musicmaker_one],  
3143 [(108, 8), (110, 8), contrabass_musicmaker_one],  
3144 [(110, 8), (112, 8), contrabass_musicmaker_one],  
3145 [(112, 8), (114, 8), contrabass_musicmaker_two],  
3146 [(114, 8), (116, 8), contrabass_musicmaker_one],  
3147 [(116, 8), (118, 8), contrabass_musicmaker_one],  
3148 [(118, 8), (120, 8), contrabass_musicmaker_two],  
3149 [(120, 8), (122, 8), contrabass_musicmaker_one],  
3150 [(122, 8), (124, 8), contrabass_musicmaker_one],  
3151 [(124, 8), (126, 8), contrabass_musicmaker_one],  
3152 [(126, 8), (128, 8), contrabass_musicmaker_one],  
3153 [(128, 8), (130, 8), contrabass_musicmaker_two],  
3154 [(130, 8), (132, 8), contrabass_musicmaker_one],  
3155 [(132, 8), (134, 8), contrabass_musicmaker_one],  
3156 [(134, 8), (136, 8), contrabass_musicmaker_one],  
3157 [(136, 8), (138, 8), contrabass_musicmaker_two],  
3158 [(138, 8), (140, 8), contrabass_musicmaker_two],  
3159 [(140, 8), (142, 8), contrabass_musicmaker_one],  
3160 [(142, 8), (144, 8), contrabass_musicmaker_one],  
3161 [(144, 8), (146, 8), contrabass_musicmaker_one],  
3162 [(146, 8), (148, 8), contrabass_musicmaker_one],  
3163 [(148, 8), (150, 8), contrabass_musicmaker_two],  
3164 [(150, 8), (152, 8), contrabass_musicmaker_two],  
3165 [(152, 8), (154, 8), contrabass_musicmaker_one],  
3166 [(154, 8), (156, 8), contrabass_musicmaker_two],  
3167 [(156, 8), (158, 8), contrabass_musicmaker_one],  
3168 [(158, 8), (160, 8), contrabass_musicmaker_one],  
3169 [(160, 8), (162, 8), contrabass_musicmaker_two],  
3170 [(162, 8), (164, 8), contrabass_musicmaker_two],  
3171 [(164, 8), (168, 8), contrabass_musicmaker_one],
```

```

3172     [(168, 8), (170, 8), contrabass_musicmaker_one],
3173     [(170, 8), (172, 8), contrabass_musicmaker_two],
3174     [(172, 8), (174, 8), contrabass_musicmaker_two],
3175     [(174, 8), (176, 8), contrabass_musicmaker_one],
3176     [(176, 8), (178, 8), contrabass_musicmaker_one],
3177     [(178, 8), (180, 8), contrabass_musicmaker_one],
3178     [(180, 8), (182, 8), contrabass_musicmaker_one],
3179     [(182, 8), (184, 8), contrabass_musicmaker_two],
3180     [(184, 8), (186, 8), contrabass_musicmaker_one],
3181     [(186, 8), (188, 8), contrabass_musicmaker_two],
3182     [(188, 8), (190, 8), contrabass_musicmaker_one],
3183     [(190, 8), (192, 8), contrabass_musicmaker_two],
3184     [(192, 8), (194, 8), contrabass_musicmaker_two],
3185     [(194, 8), (196, 8), contrabass_musicmaker_one],
3186     [(196, 8), (198, 8), contrabass_musicmaker_one],
3187     [(198, 8), (199, 8), contrabass_musicmaker_one],
3188     [(199, 8), (200, 8), contrabass_musicmaker_one],
3189   ]
3190 ])
3191
3192 all_timespan_lists = {
3193   'Voice 1': voice_1_timespan_list,
3194   'Voice 2': voice_2_timespan_list,
3195   'Voice 3': voice_3_timespan_list,
3196   'Voice 4': voice_4_timespan_list,
3197   'Voice 5': voice_5_timespan_list,
3198   'Voice 6': voice_6_timespan_list,
3199   'Voice 7': voice_7_timespan_list,
3200   'Voice 8': voice_8_timespan_list,
3201   'Voice 9': voice_9_timespan_list,
3202   'Voice 10': voice_10_timespan_list,
3203   'Voice 11': voice_11_timespan_list,
3204   'Voice 12': voice_12_timespan_list,
3205   'Voice 13': voice_13_timespan_list,
3206   'Voice 14': voice_14_timespan_list,
3207   'Voice 15': voice_15_timespan_list,
3208   'Voice 16': voice_16_timespan_list,
3209   'Voice 17': voice_17_timespan_list,
3210   'Voice 18': voice_18_timespan_list,
3211   'Voice 19': voice_19_timespan_list,
3212   'Voice 20': voice_20_timespan_list,
3213   'Voice 21': voice_21_timespan_list,
3214 }
3215
3216 global_timespan = abjad.Timespan(
3217   start_offset=0,
3218   stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
3219 )
3220
3221 for voice_name, timespan_list in all_timespan_lists.items():
3222   silences = abjad.TimespanList([global_timespan])
3223   silences.extend(timespan_list)
3224   silences.sort()
3225   silences.compute_logical_xor()
3226   for silence_timespan in silences:
3227     timespan_list.append(
3228       abjad.AnnotatedTimespan(
3229         start_offset=silence_timespan.start_offset,
3230         stop_offset=silence_timespan.stop_offset,

```

```

3231         annotation=MusicSpecifier(
3232             music_maker=None,
3233             voice_name=voice_name,
3234         ),
3235     ),
3236 )
3237 timespan_list.sort()
3238
3239 for voice_name, timespan_list in all_timespan_lists.items():
3240     shards = timespan_list.split_at_offsets(bounds)
3241     split_timespan_list = abjad.TimespanList()
3242     for shard in shards:
3243         split_timespan_list.extend(shard)
3244     split_timespan_list.sort()
3245     all_timespan_lists[voice_name] = timespan_list
3246
3247 score = abjad.Score([
3248     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
3249     abjad.StaffGroup(
3250         [
3251             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
3252             lilypond_type='Staff',),
3253             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
3254             lilypond_type='Staff',),
3255             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
3256             lilypond_type='Staff',),
3257             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
3258             lilypond_type='Staff',),
3259             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
3260             lilypond_type='Staff',),
3261             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
3262             lilypond_type='Staff',),
3263             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
3264             lilypond_type='Staff',),
3265             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
3266             lilypond_type='Staff',),
3267             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
3268             lilypond_type='Staff',),
3269             abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
3270             lilypond_type='Staff',),
3271             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
3272             lilypond_type='Staff',),
3273             abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
3274             lilypond_type='Staff',),
3275             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
3276             lilypond_type='Staff',),
3277             abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
3278             lilypond_type='Staff',),
3279             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
3280             lilypond_type='Staff',),
3281             abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
3282             lilypond_type='Staff',),
3283             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
3284             lilypond_type='Staff',),
3285             abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
3286             lilypond_type='Staff',),
3287             abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
3288             lilypond_type='Staff',),
3289             abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
3290             lilypond_type='Staff',),
3291         ]
3292     )
3293 
```

```

3271     lilypond_type='Staff',),
3272         abjad.Staff([abjad.Voice(name='Voice 21')],name='Staff 21',
3273         lilypond_type='Staff',),
3274             ],
3275             name='Staff Group',
3276         )
3277     ])
3278
3279     for time_signature in time_signatures:
3280         skip = abjad.Skip(1, multiplier=(time_signature))
3281         abjad.attach(time_signature, skip)
3282         score['Global Context'].append(skip)
3283
3284     print('Making containers ...')
3285
3286     def make_container(music_maker, durations):
3287         selections = music_maker(durations)
3288         container = abjad.Container([])
3289         container.extend(selections)
3290         return container
3291
3292
3293     def key_function(timespan):
3294         return timespan.annotation.music_maker or silence_maker
3295
3296     for voice_name, timespan_list in all_timespan_lists.items():
3297         for music_maker, grouper in itertools.groupby(
3298             timespan_list,
3299             key=key_function,
3300         ):
3301             durations = [timespan.duration for timespan in grouper]
3302             container = make_container(music_maker, durations)
3303             voice = score[voice_name]
3304             voice.append(container)
3305
3306     print('Splitting and rewriting ...')
3307     for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
3308         for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
3309             time_signature = time_signatures[i]
3310             abjad.mutate(shard).rewrite_meter(time_signature)
3311
3312     trill = TrillHandler()
3313     trill(score)
3314
3315     print('Beaming runs ...')
3316     for voice in abjad.select(score).components(abjad.Voice):
3317         for run in abjad.select(voice).runs():
3318             specifier = abjadext.rmakers.BeamSpecifier(
3319                 beam_each_division=False,
3320             )
3321             specifier(run)
3322             abjad.beam(voice[:], beam_lone_notes=False, beam_rests=False, )
3323
3324     print('Stopping Hairpins ...')
3325     for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
3326         for rest in abjad.iterate(staff).components(abjad.Rest):
3327             previous_leaf = abjad.inspect(rest).leaf(-1)
3328             if isinstance(previous_leaf, abjad.Note):
3329                 abjad.attach(abjad.StopHairpin(), rest)
3330             elif isinstance(previous_leaf, abjad.Chord):

```

```

3328         abjad.attach(abjad.StopHairpin(), rest)
3329     elif isinstance(previous_leaf, abjad.Rest):
3330         pass
3331
3332 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
3333     first_leaf = abjad.select(staff).leaves()[0]
3334     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
3335     stop_trill = abjad.LilyPondLiteral(r'\stopTrillSpan', format_slot='after')
3336     abjad.attach(stop, first_leaf)
3337     abjad.attach(stop_trill, first_leaf)
3338
3339 print('Adding attachments ...')
3340 bar_line = abjad.LilyPondLiteral(r'\bar "||"', format_slot='after')
3341 markup = abjad.Markup(r'\bold { F }')
3342 mark = abjad.RehearsalMark(markup=markup)
3343
3344 instruments = cyc([
3345     abjad.SopranoSaxophone(),
3346     abjad.SopranoSaxophone(),
3347     abjad.SopranoSaxophone(),
3348     abjad.SopranoSaxophone(),
3349     abjad.AltoSaxophone(),
3350     abjad.AltoSaxophone(),
3351     abjad.AltoSaxophone(),
3352     abjad.AltoSaxophone(),
3353     abjad.AltoSaxophone(),
3354     abjad.AltoSaxophone(),
3355     abjad.TenorSaxophone(),
3356     abjad.TenorSaxophone(),
3357     abjad.TenorSaxophone(),
3358     abjad.TenorSaxophone(),
3359     abjad.TenorSaxophone(),
3360     abjad.BaritoneSaxophone(),
3361     abjad.BaritoneSaxophone(),
3362     abjad.BaritoneSaxophone(),
3363     abjad.BassSaxophone(),
3364     abjad.BassSaxophone(),
3365     abjad.ContrabassSaxophone(),
3366 ])
3367
3368 abbreviations = cyc([
3369     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
3370     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
3371     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
3372     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
3373     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
3374     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
3375     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
3376     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
3377     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
3378     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
3379     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
3380     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
3381     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
3382     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
3383     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
3384     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
3385     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
3386     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),

```

```

3387 abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
3388 abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
3389 abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
3390 ])
3391
3392 names = cyc([
3393     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
3394     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
3395     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
3396     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
3397     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
3398     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
3399     abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
3400     abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
3401     abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
3402     abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
3403     abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
3404     abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
3405     abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
3406     abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
3407     abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
3408     abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
3409     abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
3410     abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
3411     abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
3412     abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
3413 ])
3414 ]
3415
3416 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
3417     leaf1 = abjad.select(staff).leaves()[0]
3418     abjad.attach(next(instruments), leaf1)
3419     abjad.attach(next(abbreviations), leaf1)
3420     abjad.attach(next(names), leaf1)
3421
3422 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
3423     leaf1 = abjad.select(staff).leaves()[0]
3424     last_leaf = abjad.select(staff).leaves()[-1]
3425     abjad.attach(bar_line, last_leaf)
3426
3427 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
3428     leaf1 = abjad.select(staff).leaves()[0]
3429     abjad.attach(mark, leaf1)
3430
3431 score_file = abjad.LilyPondFile.new(
3432     score,
3433     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3434     _stylesheets/abjad.ily'],
3435 )
3436
3437 abjad.SegmentMaker.comment_measure_numbers(score)
3438 ######
3439 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_F'
3440 pdf_path = f'{directory}/Section_F.pdf'
3441 path = pathlib.Path('Section_F.pdf')
3442 if path.exists():
3443     print(f'Removing {pdf_path} ...')
3444     path.unlink()

```

```

3445 time_1 = time.time()
3446 print(f'Persisting {pdf_path} ...')
3447 result = abjad.persist(score_file).as_pdf(pdf_path)
3448 print(result[0])
3449 print(result[1])
3450 print(result[2])
3451 success = result[3]
3452 if success is False:
3453     print('LilyPond failed!')
3454 time_2 = time.time()
3455 total_time = time_2 - time_1
3456 print(f'Total time: {total_time} seconds')
3457 if path.exists():
3458     print(f'Opening {pdf_path} ...')
3459     os.system(f'open {pdf_path}')
3460 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_F/Section_F.ly').readlines()
3461 open('/Users/evansdsg2/Scores/guerrero/Build/Section_F.ly', 'w').writelines(score_lines[15:-1])
3462
3463 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
3464     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3465     signature_copy = abjad.mutate(signatures).copy()
3466     staff_copy = abjad.mutate(staff).copy()
3467     part = abjad.Score()
3468     part.insert(0, staff)
3469     part.insert(0, signature_copy)
3470     part_file = abjad.LilyPondFile.new(
3471         part,
3472         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/_stylesheets/abjad.ily'],
3473         )
3474     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano'
3475     pdf_path = f'{directory}/Section_F.pdf'
3476     path = pathlib.Path('Section_F.pdf')
3477     if path.exists():
3478         print(f'Removing {pdf_path} ...')
3479         path.unlink()
3480     time_1 = time.time()
3481     print(f'Persisting {pdf_path} ...')
3482     result = abjad.persist(part_file).as_pdf(pdf_path)
3483     print(result[0])
3484     print(result[1])
3485     print(result[2])
3486     success = result[3]
3487     if success is False:
3488         print('LilyPond failed!')
3489     time_2 = time.time()
3490     total_time = time_2 - time_1
3491     print(f'Total time: {total_time} seconds')
3492     if path.exists():
3493         print(f'Opening {pdf_path} ...')
3494         os.system(f'open {pdf_path}')
3495     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano'
3496     open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano/Section_F.ly',
3497           'w').writelines(part_lines[15:-1])
3498 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):

```

```

3499 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3500 signature_copy = abjad.mutate(signatures).copy()
3501 staff_copy = abjad.mutate(staff).copy()
3502 part = abjad.Score()
3503 part.insert(0, staff)
3504 part.insert(0, signature_copy)
3505 part_file = abjad.LilyPondFile.new(
3506     part,
3507     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3508 _stylesheets/abjad.ily'],
3509     )
3509 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1'
3510 pdf_path = f'{directory}/Section_F.pdf'
3511 path = pathlib.Path('Section_F.pdf')
3512 if path.exists():
3513     print(f'Removing {pdf_path} ...')
3514     path.unlink()
3515 time_1 = time.time()
3516 print(f'Persisting {pdf_path} ...')
3517 result = abjad.persist(part_file).as_pdf(pdf_path)
3518 print(result[0])
3519 print(result[1])
3520 print(result[2])
3521 success = result[3]
3522 if success is False:
3523     print('LilyPond failed!')
3524 time_2 = time.time()
3525 total_time = time_2 - time_1
3526 print(f'Total time: {total_time} seconds')
3527 if path.exists():
3528     print(f'Opening {pdf_path} ...')
3529     os.system(f'open {pdf_path}')
3530 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/
3531 Section_F.ly').readlines()
3531 open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/Section_F.ly',
3532 'w').writelines(part_lines[15:-1])
3532
3533 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
3534     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3535     signature_copy = abjad.mutate(signatures).copy()
3536     staff_copy = abjad.mutate(staff).copy()
3537     part = abjad.Score()
3538     part.insert(0, staff)
3539     part.insert(0, signature_copy)
3540     part_file = abjad.LilyPondFile.new(
3541         part,
3542         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3543 _stylesheets/abjad.ily'],
3544         )
3544 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
3545 pdf_path = f'{directory}/Section_F.pdf'
3546 path = pathlib.Path('Section_F.pdf')
3547 if path.exists():
3548     print(f'Removing {pdf_path} ...')
3549     path.unlink()
3550 time_1 = time.time()
3551 print(f'Persisting {pdf_path} ...')
3552 result = abjad.persist(part_file).as_pdf(pdf_path)
3553 print(result[0])

```

```

3554     print(result[1])
3555     print(result[2])
3556     success = result[3]
3557     if success is False:
3558         print('LilyPond failed!')
3559     time_2 = time.time()
3560     total_time = time_2 - time_1
3561     print(f'Total time: {total_time} seconds')
3562     if path.exists():
3563         print(f'Opening {pdf_path} ...')
3564         os.system(f'open {pdf_path}')
3565     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/
3566     Section_F.ly').readlines()
3567     open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/Section_F.ly',
3568     'w').writelines(part_lines[15:-1])
3569
3570 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
3571     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3572     signature_copy = abjad.mutate(signatures).copy()
3573     staff_copy = abjad.mutate(staff).copy()
3574     part = abjad.Score()
3575     part.insert(0, staff)
3576     part.insert(0, signature_copy)
3577     part_file = abjad.LilyPondFile.new(
3578         part,
3579         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3580         _stylesheets/abjad.ily'],
3581         )
3582     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3'
3583     pdf_path = f'{directory}/Section_F.pdf'
3584     path = pathlib.Path('Section_F.pdf')
3585     if path.exists():
3586         print(f'Removing {pdf_path} ...')
3587         path.unlink()
3588     time_1 = time.time()
3589     print(f'Persisting {pdf_path} ...')
3590     result = abjad.persist(part_file).as_pdf(pdf_path)
3591     print(result[0])
3592     print(result[1])
3593     print(result[2])
3594     success = result[3]
3595     if success is False:
3596         print('LilyPond failed!')
3597     time_2 = time.time()
3598     total_time = time_2 - time_1
3599     print(f'Total time: {total_time} seconds')
3600     if path.exists():
3601         print(f'Opening {pdf_path} ...')
3602         os.system(f'open {pdf_path}')
3603     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/
3604     Section_F.ly').readlines()
3605     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/Section_F.ly',
3606     'w').writelines(part_lines[15:-1])
3607
3608 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
3609     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3610     signature_copy = abjad.mutate(signatures).copy()
3611     staff_copy = abjad.mutate(staff).copy()
3612     part = abjad.Score()

```

```

3608     part.insert(0, staff)
3609     part.insert(0, signature_copy)
3610     part_file = abjad.LilyPondFile.new(
3611         part,
3612         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3613         _stylesheets/abjad.ily'],
3614         )
3614     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1'
3615     pdf_path = f'{directory}/Section_F.pdf'
3616     path = pathlib.Path('Section_F.pdf')
3617     if path.exists():
3618         print(f'Removing {pdf_path} ...')
3619         path.unlink()
3620     time_1 = time.time()
3621     print(f'Persisting {pdf_path} ...')
3622     result = abjad.persist(part_file).as_pdf(pdf_path)
3623     print(result[0])
3624     print(result[1])
3625     print(result[2])
3626     success = result[3]
3627     if success is False:
3628         print('LilyPond failed!')
3629     time_2 = time.time()
3630     total_time = time_2 - time_1
3631     print(f'Total time: {total_time} seconds')
3632     if path.exists():
3633         print(f'Opening {pdf_path} ...')
3634         os.system(f'open {pdf_path}')
3635     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/
3635     Section_F.ly').readlines()
3636     open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/Section_F.ly', 'w'
3636     ).writelines(part_lines[15:-1])
3637
3638 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
3639     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3640     signature_copy = abjad.mutate(signatures).copy()
3641     staff_copy = abjad.mutate(staff).copy()
3642     part = abjad.Score()
3643     part.insert(0, staff)
3644     part.insert(0, signature_copy)
3645     part_file = abjad.LilyPondFile.new(
3646         part,
3647         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3647         _stylesheets/abjad.ily'],
3648         )
3649     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
3650     pdf_path = f'{directory}/Section_F.pdf'
3651     path = pathlib.Path('Section_F.pdf')
3652     if path.exists():
3653         print(f'Removing {pdf_path} ...')
3654         path.unlink()
3655     time_1 = time.time()
3656     print(f'Persisting {pdf_path} ...')
3657     result = abjad.persist(part_file).as_pdf(pdf_path)
3658     print(result[0])
3659     print(result[1])
3660     print(result[2])
3661     success = result[3]
3662     if success is False:

```

```

3663     print('LilyPond failed!')
3664 time_2 = time.time()
3665 total_time = time_2 - time_1
3666 print(f'Total time: {total_time} seconds')
3667 if path.exists():
3668     print(f'Opening {pdf_path} ...')
3669     os.system(f'open {pdf_path}')
3670 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.) alto2/
Section_F.ly').readlines()
3671 open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.) alto2/Section_F.ly', 'w'
).writelines(part_lines[15:-1])
3672
3673 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
3674     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3675     signature_copy = abjad.mutate(signatures).copy()
3676     staff_copy = abjad.mutate(staff).copy()
3677     part = abjad.Score()
3678     part.insert(0, staff)
3679     part.insert(0, signature_copy)
3680     part_file = abjad.LilyPondFile.new(
3681         part,
3682         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
3683         )
3684     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.) alto3'
3685     pdf_path = f'{directory}/Section_F.pdf'
3686     path = pathlib.Path('Section_F.pdf')
3687     if path.exists():
3688         print(f'Removing {pdf_path} ...')
3689         path.unlink()
3690     time_1 = time.time()
3691     print(f'Persisting {pdf_path} ...')
3692     result = abjad.persist(part_file).as_pdf(pdf_path)
3693     print(result[0])
3694     print(result[1])
3695     print(result[2])
3696     success = result[3]
3697     if success is False:
3698         print('LilyPond failed!')
3699     time_2 = time.time()
3700     total_time = time_2 - time_1
3701     print(f'Total time: {total_time} seconds')
3702     if path.exists():
3703         print(f'Opening {pdf_path} ...')
3704         os.system(f'open {pdf_path}')
3705     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.) alto3/
Section_F.ly').readlines()
3706     open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.) alto3/Section_F.ly', 'w'
).writelines(part_lines[15:-1])
3707
3708 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
3709     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3710     signature_copy = abjad.mutate(signatures).copy()
3711     staff_copy = abjad.mutate(staff).copy()
3712     part = abjad.Score()
3713     part.insert(0, staff)
3714     part.insert(0, signature_copy)
3715     part_file = abjad.LilyPondFile.new(
3716         part,

```

```

3717     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3718 _stylesheets/abjad.ily'],
3719     )
3720 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
3721 pdf_path = f'{directory}/Section_F.pdf'
3722 path = pathlib.Path('Section_F.pdf')
3723 if path.exists():
3724     print(f'Removing {pdf_path} ...')
3725     path.unlink()
3726 time_1 = time.time()
3727 print(f'Persisting {pdf_path} ...')
3728 result = abjad.persist(part_file).as_pdf(pdf_path)
3729 print(result[0])
3730 print(result[1])
3731 print(result[2])
3732 success = result[3]
3733 if success is False:
3734     print('LilyPond failed!')
3735 time_2 = time.time()
3736 total_time = time_2 - time_1
3737 print(f'Total time: {total_time} seconds')
3738 if path.exists():
3739     print(f'Opening {pdf_path} ...')
3740     os.system(f'open {pdf_path}')
3741 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/
3742 Section_F.ly').readlines()
3743 open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/Section_F.ly', 'w'
3744 ).writelines(part_lines[15:-1])
3745
3746 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
3747     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3748     signature_copy = abjad.mutate(signatures).copy()
3749     staff_copy = abjad.mutate(staff).copy()
3750     part = abjad.Score()
3751     part.insert(0, staff)
3752     part.insert(0, signature_copy)
3753     part_file = abjad.LilyPondFile.new(
3754         part,
3755         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3756 _stylesheets/abjad.ily'],
3757         )
3758 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
3759 pdf_path = f'{directory}/Section_F.pdf'
3760 path = pathlib.Path('Section_F.pdf')
3761 if path.exists():
3762     print(f'Removing {pdf_path} ...')
3763     path.unlink()
3764 time_1 = time.time()
3765 print(f'Persisting {pdf_path} ...')
3766 result = abjad.persist(part_file).as_pdf(pdf_path)
3767 print(result[0])
3768 print(result[1])
3769 print(result[2])
3770 success = result[3]
3771 if success is False:
3772     print('LilyPond failed!')
3773 time_2 = time.time()
3774 total_time = time_2 - time_1
3775 print(f'Total time: {total_time} seconds')

```

```

3772     if path.exists():
3773         print(f'Opening {pdf_path} ...')
3774         os.system(f'open {pdf_path}')
3775     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/
3776     Section_F.ly').readlines()
3777     open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/Section_F.ly', 'w'
3778     ).writelines(part_lines[15:-1])
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
if path.exists():
    print(f'Opening {pdf_path} ...')
    os.system(f'open {pdf_path}')
part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/
Section_F.ly').readlines()
open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/Section_F.ly', 'w'
).writelines(part_lines[15:-1])

for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
signatures = abjad.select(score['Global Context']).components(abjad.Staff)
signature_copy = abjad.mutate(signatures).copy()
staff_copy = abjad.mutate(staff).copy()
part = abjad.Score()
part.insert(0, staff)
part.insert(0, signature_copy)
part_file = abjad.LilyPondFile.new(
    part,
    includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
)
directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.')alto6'
pdf_path = f'{directory}/Section_F.pdf'
path = pathlib.Path('Section_F.pdf')
if path.exists():
    print(f'Removing {pdf_path} ...')
    path.unlink()
time_1 = time.time()
print(f'Persisting {pdf_path} ...')
result = abjad.persist(part_file).as_pdf(pdf_path)
print(result[0])
print(result[1])
print(result[2])
success = result[3]
if success is False:
    print('LilyPond failed!')
time_2 = time.time()
total_time = time_2 - time_1
print(f'Total time: {total_time} seconds')
if path.exists():
    print(f'Opening {pdf_path} ...')
    os.system(f'open {pdf_path}')
part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.')alto6/
Section_F.ly').readlines()
open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.')alto6/Section_F.ly', 'w'
).writelines(part_lines[15:-1])

for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
signatures = abjad.select(score['Global Context']).components(abjad.Staff)
signature_copy = abjad.mutate(signatures).copy()
staff_copy = abjad.mutate(staff).copy()
part = abjad.Score()
part.insert(0, staff)
part.insert(0, signature_copy)
part_file = abjad.LilyPondFile.new(
    part,
    includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
)
directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.')tenor1'

```

```

3825 pdf_path = f'{directory}/Section_F.pdf'
3826 path = pathlib.Path('Section_F.pdf')
3827 if path.exists():
3828     print(f'Removing {pdf_path} ...')
3829     path.unlink()
3830 time_1 = time.time()
3831 print(f'Persisting {pdf_path} ...')
3832 result = abjad.persist(part_file).as_pdf(pdf_path)
3833 print(result[0])
3834 print(result[1])
3835 print(result[2])
3836 success = result[3]
3837 if success is False:
3838     print('LilyPond failed!')
3839 time_2 = time.time()
3840 total_time = time_2 - time_1
3841 print(f'Total time: {total_time} seconds')
3842 if path.exists():
3843     print(f'Opening {pdf_path} ...')
3844     os.system(f'open {pdf_path}')
3845 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
3846 Section_F.ly').readlines()
3847 open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Section_F.ly', ,
3848 w').writelines(part_lines[15:-1])
3849
3850 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
3851     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3852     signature_copy = abjad.mutate(signatures).copy()
3853     staff_copy = abjad.mutate(staff).copy()
3854     part = abjad.Score()
3855     part.insert(0, staff)
3856     part.insert(0, signature_copy)
3857     part_file = abjad.LilyPondFile.new(
3858         part,
3859         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3860         _stylesheets/abjad.ily'],
3861         )
3862 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
3863 pdf_path = f'{directory}/Section_F.pdf'
3864 path = pathlib.Path('Section_F.pdf')
3865 if path.exists():
3866     print(f'Removing {pdf_path} ...')
3867     path.unlink()
3868 time_1 = time.time()
3869 print(f'Persisting {pdf_path} ...')
3870 result = abjad.persist(part_file).as_pdf(pdf_path)
3871 print(result[0])
3872 print(result[1])
3873 print(result[2])
3874 success = result[3]
3875 if success is False:
3876     print('LilyPond failed!')
3877 time_2 = time.time()
3878 total_time = time_2 - time_1
3879 print(f'Total time: {total_time} seconds')
3880 if path.exists():
3881     print(f'Opening {pdf_path} ...')
3882     os.system(f'open {pdf_path}')
3883 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/

```

```

Section_F.ly').readlines()
3881 open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/Section_F.ly', 'w').writelines(part_lines[15:-1])

3882
3883 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
3884     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3885     signature_copy = abjad.mutate(signatures).copy()
3886     staff_copy = abjad.mutate(staff).copy()
3887     part = abjad.Score()
3888     part.insert(0, staff)
3889     part.insert(0, signature_copy)
3890     part_file = abjad.LilyPondFile.new(
3891         part,
3892         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3893         _stylesheets/abjad.ily'],
3894         )
3895     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3'
3896     pdf_path = f'{directory}/Section_F.pdf'
3897     path = pathlib.Path('Section_F.pdf')
3898     if path.exists():
3899         print(f'Removing {pdf_path} ...')
3900         path.unlink()
3901     time_1 = time.time()
3902     print(f'Persisting {pdf_path} ...')
3903     result = abjad.persist(part_file).as_pdf(pdf_path)
3904     print(result[0])
3905     print(result[1])
3906     print(result[2])
3907     success = result[3]
3908     if success is False:
3909         print('LilyPond failed!')
3910     time_2 = time.time()
3911     total_time = time_2 - time_1
3912     print(f'Total time: {total_time} seconds')
3913     if path.exists():
3914         print(f'Opening {pdf_path} ...')
3915         os.system(f'open {pdf_path}')
3916     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3/
3917     Section_F.ly').readlines()
3918     open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3/Section_F.ly', 'w').writelines(part_lines[15:-1])

3919
3920 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
3921     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3922     signature_copy = abjad.mutate(signatures).copy()
3923     staff_copy = abjad.mutate(staff).copy()
3924     part = abjad.Score()
3925     part.insert(0, staff)
3926     part.insert(0, signature_copy)
3927     part_file = abjad.LilyPondFile.new(
3928         part,
3929         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3930         _stylesheets/abjad.ily'],
3931         )
3932     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.)tenor4'
3933     pdf_path = f'{directory}/Section_F.pdf'
3934     path = pathlib.Path('Section_F.pdf')
3935     if path.exists():
3936         print(f'Removing {pdf_path} ...')

```

```

3934     path.unlink()
3935 time_1 = time.time()
3936 print(f'Persisting {pdf_path} ...')
3937 result = abjad.persist(part_file).as_pdf(pdf_path)
3938 print(result[0])
3939 print(result[1])
3940 print(result[2])
3941 success = result[3]
3942 if success is False:
3943     print('LilyPond failed!')
3944 time_2 = time.time()
3945 total_time = time_2 - time_1
3946 print(f'Total time: {total_time} seconds')
3947 if path.exists():
3948     print(f'Opening {pdf_path} ...')
3949     os.system(f'open {pdf_path}')
3950 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
Section_F.ly').readlines()
3951 open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_F.ly', ,
w').writelines(part_lines[15:-1])
3952
3953 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
3954 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3955 signature_copy = abjad.mutate(signatures).copy()
3956 staff_copy = abjad.mutate(staff).copy()
3957 part = abjad.Score()
3958 part.insert(0, staff)
3959 part.insert(0, signature_copy)
3960 part_file = abjad.LilyPondFile.new(
3961     part,
3962     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
3963     )
3964 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5'
3965 pdf_path = f'{directory}/Section_F.pdf'
3966 path = pathlib.Path('Section_F.pdf')
3967 if path.exists():
3968     print(f'Removing {pdf_path} ...')
3969     path.unlink()
3970 time_1 = time.time()
3971 print(f'Persisting {pdf_path} ...')
3972 result = abjad.persist(part_file).as_pdf(pdf_path)
3973 print(result[0])
3974 print(result[1])
3975 print(result[2])
3976 success = result[3]
3977 if success is False:
3978     print('LilyPond failed!')
3979 time_2 = time.time()
3980 total_time = time_2 - time_1
3981 print(f'Total time: {total_time} seconds')
3982 if path.exists():
3983     print(f'Opening {pdf_path} ...')
3984     os.system(f'open {pdf_path}')
3985 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/
Section_F.ly').readlines()
3986 open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/Section_F.ly', ,
w').writelines(part_lines[15:-1])
3987

```

```

3988 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
3989     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3990     signature_copy = abjad.mutate(signatures).copy()
3991     staff_copy = abjad.mutate(staff).copy()
3992     part = abjad.Score()
3993     part.insert(0, staff)
3994     part.insert(0, signature_copy)
3995     part_file = abjad.LilyPondFile.new(
3996         part,
3997         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3998         _stylesheets/abjad.ily'],
3999         )
4000     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1'
4001     pdf_path = f'{directory}/Section_F.pdf'
4002     path = pathlib.Path('Section_F.pdf')
4003     if path.exists():
4004         print(f'Removing {pdf_path} ...')
4005         path.unlink()
4006     time_1 = time.time()
4007     print(f'Persisting {pdf_path} ...')
4008     result = abjad.persist(part_file).as_pdf(pdf_path)
4009     print(result[0])
4010     print(result[1])
4011     print(result[2])
4012     success = result[3]
4013     if success is False:
4014         print('LilyPond failed!')
4015     time_2 = time.time()
4016     total_time = time_2 - time_1
4017     print(f'Total time: {total_time} seconds')
4018     if path.exists():
4019         print(f'Opening {pdf_path} ...')
4020         os.system(f'open {pdf_path}')
4021     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/
4022     Section_F.ly').readlines()
4023     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/Section_F.ly',
4024     'w').writelines(part_lines[15:-1])

4025 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
4026     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4027     signature_copy = abjad.mutate(signatures).copy()
4028     staff_copy = abjad.mutate(staff).copy()
4029     part = abjad.Score()
4030     part.insert(0, staff)
4031     part.insert(0, signature_copy)
4032     part_file = abjad.LilyPondFile.new(
4033         part,
4034         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
4035         _stylesheets/abjad.ily'],
4036         )
4037     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2'
4038     pdf_path = f'{directory}/Section_F.pdf'
4039     path = pathlib.Path('Section_F.pdf')
4040     if path.exists():
4041         print(f'Removing {pdf_path} ...')
4042         path.unlink()

```

```

4043     print(result[0])
4044     print(result[1])
4045     print(result[2])
4046     success = result[3]
4047     if success is False:
4048         print('LilyPond failed!')
4049     time_2 = time.time()
4050     total_time = time_2 - time_1
4051     print(f'Total time: {total_time} seconds')
4052     if path.exists():
4053         print(f'Opening {pdf_path} ...')
4054         os.system(f'open {pdf_path}')
4055     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/
Section_F.ly').readlines()
4056     open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/Section_F.ly',
        'w').writelines(part_lines[15:-1])
4057
4058 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
4059     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4060     signature_copy = abjad.mutate(signatures).copy()
4061     staff_copy = abjad.mutate(staff).copy()
4062     part = abjad.Score()
4063     part.insert(0, staff)
4064     part.insert(0, signature_copy)
4065     part_file = abjad.LilyPondFile.new(
4066         part,
4067         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
4068         )
4069     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3'
4070     pdf_path = f'{directory}/Section_F.pdf'
4071     path = pathlib.Path('Section_F.pdf')
4072     if path.exists():
4073         print(f'Removing {pdf_path} ...')
4074         path.unlink()
4075     time_1 = time.time()
4076     print(f'Persisting {pdf_path} ...')
4077     result = abjad.persist(part_file).as_pdf(pdf_path)
4078     print(result[0])
4079     print(result[1])
4080     print(result[2])
4081     success = result[3]
4082     if success is False:
4083         print('LilyPond failed!')
4084     time_2 = time.time()
4085     total_time = time_2 - time_1
4086     print(f'Total time: {total_time} seconds')
4087     if path.exists():
4088         print(f'Opening {pdf_path} ...')
4089         os.system(f'open {pdf_path}')
4090     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/
Section_F.ly').readlines()
4091     open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3/Section_F.ly',
        'w').writelines(part_lines[15:-1])
4092
4093 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
4094     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4095     signature_copy = abjad.mutate(signatures).copy()
4096     staff_copy = abjad.mutate(staff).copy()

```

```

4097 part = abjad.Score()
4098 part.insert(0, staff)
4099 part.insert(0, signature_copy)
4100 part_file = abjad.LilyPondFile.new(
4101     part,
4102     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
4103 _stylesheets/abjad.ily'],
4104     )
4105 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
4106 pdf_path = f'{directory}/Section_F.pdf'
4107 path = pathlib.Path('Section_F.pdf')
4108 if path.exists():
4109     print(f'Removing {pdf_path} ...')
4110     path.unlink()
4111 time_1 = time.time()
4112 print(f'Persisting {pdf_path} ...')
4113 result = abjad.persist(part_file).as_pdf(pdf_path)
4114 print(result[0])
4115 print(result[1])
4116 print(result[2])
4117 success = result[3]
4118 if success is False:
4119     print('LilyPond failed!')
4120 time_2 = time.time()
4121 total_time = time_2 - time_1
4122 print(f'Total time: {total_time} seconds')
4123 if path.exists():
4124     print(f'Opening {pdf_path} ...')
4125     os.system(f'open {pdf_path}')
4126 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
4127 Section_F.ly').readlines()
4128 open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_F.ly', 'w'
4129 .writelines(part_lines[15:-1])
4130
4131 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
4132     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4133     signature_copy = abjad.mutate(signatures).copy()
4134     staff_copy = abjad.mutate(staff).copy()
4135     part = abjad.Score()
4136     part.insert(0, staff)
4137     part.insert(0, signature_copy)
4138     part_file = abjad.LilyPondFile.new(
4139         part,
4140         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
4141 _stylesheets/abjad.ily'],
4142         )
4143 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
4144 pdf_path = f'{directory}/Section_F.pdf'
4145 path = pathlib.Path('Section_F.pdf')
4146 if path.exists():
4147     print(f'Removing {pdf_path} ...')
4148     path.unlink()
4149 time_1 = time.time()
4150 print(f'Persisting {pdf_path} ...')
4151 result = abjad.persist(part_file).as_pdf(pdf_path)
4152 print(result[0])
4153 print(result[1])
4154 print(result[2])
4155 success = result[3]
```

```

4152     if success is False:
4153         print('LilyPond failed!')
4154     time_2 = time.time()
4155     total_time = time_2 - time_1
4156     print(f'Total time: {total_time} seconds')
4157     if path.exists():
4158         print(f'Opening {pdf_path} ...')
4159         os.system(f'open {pdf_path}')
4160     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
4161     Section_F.ly').readlines()
4162     open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_F.ly', 'w'
4163     ).writelines(part_lines[15:-1])
4164
4165 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
4166     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
4167     signature_copy = abjad.mutate(signatures).copy()
4168     staff_copy = abjad.mutate(staff).copy()
4169     part = abjad.Score()
4170     part.insert(0, staff)
4171     part.insert(0, signature_copy)
4172     part_file = abjad.LilyPondFile.new(
4173         part,
4174         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
4175         _stylesheets/abjad.ily'],
4176         )
4177     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
4178     pdf_path = f'{directory}/Section_F.pdf'
4179     path = pathlib.Path('Section_F.pdf')
4180     if path.exists():
4181         print(f'Removing {pdf_path} ...')
4182         path.unlink()
4183     time_1 = time.time()
4184     print(f'Persisting {pdf_path} ...')
4185     result = abjad.persist(part_file).as_pdf(pdf_path)
4186     print(result[0])
4187     print(result[1])
4188     print(result[2])
4189     success = result[3]
4190     if success is False:
4191         print('LilyPond failed!')
4192     time_2 = time.time()
4193     total_time = time_2 - time_1
4194     print(f'Total time: {total_time} seconds')
4195     if path.exists():
4196         print(f'Opening {pdf_path} ...')
4197         os.system(f'open {pdf_path}')
4198     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/
4199     Section_F.ly').readlines()
4200     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_F.ly',
4201     'w').writelines(part_lines[15:-1])

```

Listing 3.7: Invocation Source Code

## 3.8 Section G

```

1 import abjad
2 import itertools

```

```

3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11 from evans.general_tools.random_walk import randomWalk
12 from evans.general_tools.rotate import rotate
13 from evans.general_tools.mirror import mirror
14
15 print('Interpreting file ...')
16
17 time_signatures = [
18     abjad.TimeSignature(pair) for pair in [
19         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
20         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
21         (4, 4), (4, 4),
22         # (4, 4), (4, 4), (4, 4), (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
23         # (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
24     ]
25 ]
26
27 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
28
29 def cyc(lst):
30     count = 0
31     while True:
32         yield lst[count%len(lst)]
33         count += 1
34
35 def grouper(lst1, lst2):
36     def cyc(lst):
37         c = 0
38         while True:
39             yield lst[c%len(lst)]
40             c += 1
41     lst1 = cyc(lst1)
42     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2]
43
44 def reduceMod(list_length, rw):
45     return [(x % list_length) for x in rw]
46
47 # -3 at bottom of chord for completion
48 soprano_chord = [27, ]
49 soprano_1_chord = [[13.25, 16, 26.25, ], 22,]
50 soprano_2_chord = [[13, 14.75, 26.25, ], 16,] #maybe it's 13.25?
51 soprano_3_chord = [[12.75, 15.5, 26, ], 13,]
52 alto_1_chord = [[12.5, 19, 27.75, 34, ], 20,]
53 alto_2_chord = [[12.5, 15.25, 25.5, ], 12,]
54 alto_3_chord = [[1.75, 13.5, 22.25, 27, 30, ], 1,]
55 alto_4_chord = [[12.5, 15.25, 25.5, ], 20,]
56 alto_5_chord = [[1.75, 13.5, 22.25, 27, 30, ], 12,]
57 alto_6_chord = [[12.5, 19, 27.75, 34, ], 1,]
58 tenor_1_chord = [[6, 17.5, ], 17,]
59 tenor_2_chord = [[6, 17.5, 25.5, 30, ], 6,]
60 tenor_3_chord = [[6, 17.5, 25.5, 30.75, ], -1]
```

```

61 tenor_4_chord = [[6, 17.5, ], 17,]
62 tenor_5_chord = [[6, 17.5, 25.5, 30.75, ], 6,]
63 baritone_1_chord = [[13.25, 27.5, 33.75, ], 13,]
64 baritone_2_chord = [[4, 16.5, 23.5, ], 6,]
65 baritone_3_chord = [[7.75, 17.75, 25.5, 34, ], 4,]
66 bass_1_chord = [11, ]
67 bass_2_chord = [9, ]
68 contrabass_chord = [-2, 2, 7, -2, 2, 7, 2, -2]
69
70 def reduceMod(x, rw):
71     return [(y % x) for y in rw]
72
73 walk_list = []
74 for x in range(-1, 30):
75     walk_list.append(x)
76 mirrored_walk_list = mirror(walk_list, sequential_duplicates=False)
77
78 sopranino_walk_chord = rotate(mirrored_walk_list, 28)
79 sopranino_random_walk_notes = [x for x in randomWalk(
80     random_seed=101,
81     length=1000,
82     step_list=[1],
83     mapped_list=sopranino_walk_chord
84         )
85     ]
86
87 soprano_1_walk_chord = rotate(mirrored_walk_list, 23)
88 soprano_1_random_walk_notes = [x for x in randomWalk(
89     random_seed=102,
90     length=1000,
91     step_list=[1],
92     mapped_list=soprano_1_walk_chord
93         )
94     ]
95
96 soprano_2_walk_chord = rotate(mirrored_walk_list, 17)
97 soprano_2_random_walk_notes = [x for x in randomWalk(
98     random_seed=103,
99     length=1000,
100    step_list=[1],
101    mapped_list=soprano_2_walk_chord
102        )
103     ]
104
105 soprano_3_walk_chord = rotate(mirrored_walk_list, 14)
106 soprano_3_random_walk_notes = [x for x in randomWalk(
107     random_seed=104,
108     length=1000,
109     step_list=[1],
110     mapped_list=soprano_3_walk_chord
111         )
112     ]
113
114 alto_1_walk_chord = rotate(mirrored_walk_list, 21)
115 alto_1_random_walk_notes = [x for x in randomWalk(
116     random_seed=105,
117     length=1000,
118     step_list=[1],
119     mapped_list=alto_1_walk_chord

```

```
120         )
121     ]
122
123 alto_2_walk_chord = rotate(mirrored_walk_list, 13)
124 alto_2_random_walk_notes = [x for x in randomWalk(
125     random_seed=106,
126     length=1000,
127     step_list=[1],
128     mapped_list=alto_2_walk_chord
129     )
130   ]
131
132 alto_3_walk_chord = rotate(mirrored_walk_list, 2)
133 alto_3_random_walk_notes = [x for x in randomWalk(
134     random_seed=107,
135     length=1000,
136     step_list=[1],
137     mapped_list=alto_3_walk_chord
138     )
139   ]
140
141 alto_4_walk_chord = rotate(mirrored_walk_list, 21)
142 alto_4_random_walk_notes = [x for x in randomWalk(
143     random_seed=108,
144     length=1000,
145     step_list=[1],
146     mapped_list=alto_4_walk_chord
147     )
148   ]
149
150 alto_5_walk_chord = rotate(mirrored_walk_list, 13)
151 alto_5_random_walk_notes = [x for x in randomWalk(
152     random_seed=109,
153     length=1000,
154     step_list=[1],
155     mapped_list=alto_5_walk_chord
156     )
157   ]
158
159 alto_6_walk_chord = rotate(mirrored_walk_list, 2)
160 alto_6_random_walk_notes = [x for x in randomWalk(
161     random_seed=110,
162     length=1000,
163     step_list=[1],
164     mapped_list=alto_6_walk_chord
165     )
166   ]
167
168 tenor_1_walk_chord = rotate(mirrored_walk_list, 18)
169 tenor_1_random_walk_notes = [x for x in randomWalk(
170     random_seed=111,
171     length=1000,
172     step_list=[1],
173     mapped_list=tenor_1_walk_chord
174     )
175   ]
176
177 tenor_2_walk_chord = rotate(mirrored_walk_list, 7)
178 tenor_2_random_walk_notes = [x for x in randomWalk(
```

```
179     random_seed=112,
180     length=1000,
181     step_list=[1],
182     mapped_list=tenor_2_walk_chord
183     )
184   ]
185
186 tenor_3_walk_chord = rotate(mirrored_walk_list, 0)
187 tenor_3_random_walk_notes = [x for x in randomWalk(
188   random_seed=113,
189   length=1000,
190   step_list=[1],
191   mapped_list=tenor_3_walk_chord
192   )
193   ]
194
195 tenor_4_walk_chord = rotate(mirrored_walk_list, 18)
196 tenor_4_random_walk_notes = [x for x in randomWalk(
197   random_seed=114,
198   length=1000,
199   step_list=[1],
200   mapped_list=tenor_4_walk_chord
201   )
202   ]
203
204 tenor_5_walk_chord = rotate(mirrored_walk_list, 7)
205 tenor_5_random_walk_notes = [x for x in randomWalk(
206   random_seed=115,
207   length=1000,
208   step_list=[1],
209   mapped_list=tenor_5_walk_chord
210   )
211   ]
212
213 baritone_1_walk_chord = rotate(mirrored_walk_list, 14)
214 baritone_1_random_walk_notes = [x for x in randomWalk(
215   random_seed=116,
216   length=1000,
217   step_list=[1],
218   mapped_list=baritone_1_walk_chord
219   )
220   ]
221
222 baritone_2_walk_chord = rotate(mirrored_walk_list, 7)
223 baritone_2_random_walk_notes = [x for x in randomWalk(
224   random_seed=117,
225   length=1000,
226   step_list=[1],
227   mapped_list=baritone_2_walk_chord
228   )
229   ]
230
231 baritone_3_walk_chord = rotate(mirrored_walk_list, 5)
232 baritone_3_random_walk_notes = [x for x in randomWalk(
233   random_seed=118,
234   length=1000,
235   step_list=[1],
236   mapped_list=baritone_3_walk_chord
237   )
```

```

238     ]
239
240 bass_1_walk_chord = rotate(mirrored_walk_list, 12)
241 bass_1_random_walk_notes = [x for x in randomWalk(
242     random_seed=119,
243     length=1000,
244     step_list=[1],
245     mapped_list=bass_1_walk_chord
246     )
247     ]
248
249 bass_2_walk_chord = rotate(mirrored_walk_list, 10)
250 bass_2_random_walk_notes = [x for x in randomWalk(
251     random_seed=120,
252     length=1000,
253     step_list=[1],
254     mapped_list=bass_2_walk_chord
255     )
256     ]
257
258 contrabass_walk_chord = rotate(mirrored_walk_list, 3)
259 contrabass_random_walk_notes = [x for x in randomWalk(
260     random_seed=121,
261     length=1000,
262     step_list=[1],
263     mapped_list=contrabass_walk_chord
264     )
265     ]
266
267 rmaker_one = abjadext.rmakers.NoteRhythmMaker()
268
269 rmaker_two = abjadext.rmakers.TaleaRhythmMaker(
270     talea=abjadext.rmakers.Talea(
271         counts=[1, 1, 2, 2, 1, 2, 1, 3, 2, 5, 3, 2, 1, 1, 2, 1, 5, 1, 3, ],
272         denominator=8,
273         ),
274     beamSpecifier=abjadext.rmakers.BeamSpecifier(
275         beamDivisionsTogether=True,
276         beamRests=False,
277         ),
278     extraCountsPerDivision=[-1, 0, 1, -1, 1, 0, ],
279     logicalTieMasks=[
280         abjadext.rmakers.silence([8], 5),
281         ],
282     divisionMasks=[
283         abjadext.rmakers.SilenceMask(
284             pattern=abjad.index([2], 11),
285             ),
286         ],
287     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
288         trivialize=True,
289         extractTrivial=True,
290         rewriteRestFilled=True,
291         rewriteSustained=True,
292         ),
293     )
294
295 attachmentHandlerOne = AttachmentHandler(
296     startingDynamic='mf',

```

```
297     ending_dynamic='ppp',
298     hairpin='>',
299     articulation_list=['flageolet', 'halfopen', '', 'flageolet', 'flageolet', '',
300     'halfopen', '', 'halfopen', 'halfopen', '', '', '', 'flageolet', 'flageolet', '',
301     'halfopen', ],
302 )
303 attachment_handler_two = AttachmentHandler(
304     starting_dynamic='pp',
305     ending_dynamic='mp',
306     hairpin='<',
307     articulation_list=['halfopen', 'halfopen', 'halfopen', 'halfopen', 'halfopen',
308     '', 'halfopen', '', ],
309 )
310 #####sopranino#####
311 sopranino_musicmaker_one = MusicMaker(
312     rmaker=rmaker_one,
313     pitches=sopranino_chord,
314     continuous=True,
315     attachment_handler=attachment_handler_one,
316 )
317 sopranino_musicmaker_two = MusicMaker(
318     rmaker=rmaker_two,
319     pitches=sopranino_random_walk_notes,
320     continuous=True,
321     attachment_handler=attachment_handler_one,
322 )
323 sopranino_musicmaker_three = MusicMaker(
324     rmaker=rmaker_two,
325     pitches=sopranino_random_walk_notes,
326     continuous=True,
327     attachment_handler=attachment_handler_two,
328 #####soprano_one#####
329 soprano_one_musicmaker_one = MusicMaker(
330     rmaker=rmaker_one,
331     pitches=soprano_1_chord,
332     continuous=True,
333     attachment_handler=attachment_handler_one,
334 )
335 soprano_one_musicmaker_two = MusicMaker(
336     rmaker=rmaker_two,
337     pitches=soprano_1_random_walk_notes,
338     continuous=True,
339     attachment_handler=attachment_handler_one,
340 )
341 soprano_one_musicmaker_three = MusicMaker(
342     rmaker=rmaker_two,
343     pitches=soprano_1_random_walk_notes,
344     continuous=True,
345     attachment_handler=attachment_handler_two,
346 )
347 #####soprano_two#####
348 soprano_two_musicmaker_one = MusicMaker(
349     rmaker=rmaker_one,
350     pitches=soprano_2_chord,
351     continuous=True,
352     attachment_handler=attachment_handler_one,
```

```
353 )
354 soprano_two_musicmaker_two = MusicMaker(
355     rmaker=rmaker_two,
356     pitches=soprano_2_random_walk_notes,
357     continuous=True,
358     attachment_handler=attachment_handler_one,
359 )
360 soprano_two_musicmaker_three = MusicMaker(
361     rmaker=rmaker_two,
362     pitches=soprano_2_random_walk_notes,
363     continuous=True,
364     attachment_handler=attachment_handler_two,
365 )
366 #####soprano_three#####
367 soprano_three_musicmaker_one = MusicMaker(
368     rmaker=rmaker_one,
369     pitches=soprano_3_chord,
370     continuous=True,
371     attachment_handler=attachment_handler_one,
372 )
373 soprano_three_musicmaker_two = MusicMaker(
374     rmaker=rmaker_two,
375     pitches=soprano_3_random_walk_notes,
376     continuous=True,
377     attachment_handler=attachment_handler_one,
378 )
379 soprano_three_musicmaker_three = MusicMaker(
380     rmaker=rmaker_two,
381     pitches=soprano_3_random_walk_notes,
382     continuous=True,
383     attachment_handler=attachment_handler_two,
384 )
385 #####alto_one#####
386 alto_one_musicmaker_one = MusicMaker(
387     rmaker=rmaker_one,
388     pitches=alto_1_chord,
389     continuous=True,
390     attachment_handler=attachment_handler_one,
391 )
392 alto_one_musicmaker_two = MusicMaker(
393     rmaker=rmaker_two,
394     pitches=soprano_1_random_walk_notes,
395     continuous=True,
396     attachment_handler=attachment_handler_one,
397 )
398 alto_one_musicmaker_three = MusicMaker(
399     rmaker=rmaker_two,
400     pitches=soprano_1_random_walk_notes,
401     continuous=True,
402     attachment_handler=attachment_handler_two,
403 )
404 #####alto_two#####
405 alto_two_musicmaker_one = MusicMaker(
406     rmaker=rmaker_one,
407     pitches=alto_2_chord,
408     continuous=True,
409     attachment_handler=attachment_handler_one,
410 )
411 alto_two_musicmaker_two = MusicMaker(
```

```
412     rmaker=rmaker_two,
413     pitches=soprano_2_random_walk_notes,
414     continuous=True,
415     attachment_handler=attachment_handler_one,
416 )
417 alto_two_musicmaker_three = MusicMaker(
418     rmaker=rmaker_two,
419     pitches=soprano_2_random_walk_notes,
420     continuous=True,
421     attachment_handler=attachment_handler_two,
422 )
423 #####alto_three#####
424 alto_three_musicmaker_one = MusicMaker(
425     rmaker=rmaker_one,
426     pitches=alto_3_chord,
427     continuous=True,
428     attachment_handler=attachment_handler_one,
429 )
430 alto_three_musicmaker_two = MusicMaker(
431     rmaker=rmaker_two,
432     pitches=soprano_3_random_walk_notes,
433     continuous=True,
434     attachment_handler=attachment_handler_one,
435 )
436 alto_three_musicmaker_three = MusicMaker(
437     rmaker=rmaker_two,
438     pitches=soprano_3_random_walk_notes,
439     continuous=True,
440     attachment_handler=attachment_handler_two,
441 )
442 #####alto_four#####
443 alto_four_musicmaker_one = MusicMaker(
444     rmaker=rmaker_one,
445     pitches=alto_4_chord,
446     continuous=True,
447     attachment_handler=attachment_handler_one,
448 )
449 alto_four_musicmaker_two = MusicMaker(
450     rmaker=rmaker_two,
451     pitches=alto_4_random_walk_notes,
452     continuous=True,
453     attachment_handler=attachment_handler_one,
454 )
455 alto_four_musicmaker_three = MusicMaker(
456     rmaker=rmaker_two,
457     pitches=alto_4_random_walk_notes,
458     continuous=True,
459     attachment_handler=attachment_handler_two,
460 )
461 #####alto_five#####
462 alto_five_musicmaker_one = MusicMaker(
463     rmaker=rmaker_one,
464     pitches=alto_5_chord,
465     continuous=True,
466     attachment_handler=attachment_handler_one,
467 )
468 alto_five_musicmaker_two = MusicMaker(
469     rmaker=rmaker_two,
470     pitches=alto_5_random_walk_notes,
```

```
471     continuous=True,
472     attachment_handler=attachment_handler_one,
473 )
474 alto_five_musicmaker_three = MusicMaker(
475     rmaker=rmaker_two,
476     pitches=alto_5_random_walk_notes,
477     continuous=True,
478     attachment_handler=attachment_handler_two,
479 )
480 #####alto_six#####
481 alto_six_musicmaker_one = MusicMaker(
482     rmaker=rmaker_one,
483     pitches=alto_6_chord,
484     continuous=True,
485     attachment_handler=attachment_handler_one,
486 )
487 alto_six_musicmaker_two = MusicMaker(
488     rmaker=rmaker_two,
489     pitches=alto_6_random_walk_notes,
490     continuous=True,
491     attachment_handler=attachment_handler_one,
492 )
493 alto_six_musicmaker_three = MusicMaker(
494     rmaker=rmaker_two,
495     pitches=alto_6_random_walk_notes,
496     continuous=True,
497     attachment_handler=attachment_handler_two,
498 )
499 #####tenor_one#####
500 tenor_one_musicmaker_one = MusicMaker(
501     rmaker=rmaker_one,
502     pitches=tenor_1_chord,
503     continuous=True,
504     attachment_handler=attachment_handler_one,
505 )
506 tenor_one_musicmaker_two = MusicMaker(
507     rmaker=rmaker_two,
508     pitches=tenor_1_random_walk_notes,
509     continuous=True,
510     attachment_handler=attachment_handler_one,
511 )
512 tenor_one_musicmaker_three = MusicMaker(
513     rmaker=rmaker_two,
514     pitches=tenor_1_random_walk_notes,
515     continuous=True,
516     attachment_handler=attachment_handler_two,
517 )
518 #####tenor_two#####
519 tenor_two_musicmaker_one = MusicMaker(
520     rmaker=rmaker_one,
521     pitches=tenor_2_chord,
522     continuous=True,
523     attachment_handler=attachment_handler_one,
524 )
525 tenor_two_musicmaker_two = MusicMaker(
526     rmaker=rmaker_two,
527     pitches=tenor_2_random_walk_notes,
528     continuous=True,
529     attachment_handler=attachment_handler_one,
```

```
530 )
531 tenor_two_musicmaker_three = MusicMaker(
532     rmaker=rmaker_two,
533     pitches=tenor_2_random_walk_notes,
534     continuous=True,
535     attachment_handler=attachment_handler_two,
536 )
537 #####tenor_three#####
538 tenor_three_musicmaker_one = MusicMaker(
539     rmaker=rmaker_one,
540     pitches=tenor_3_chord,
541     continuous=True,
542     attachment_handler=attachment_handler_one,
543 )
544 tenor_three_musicmaker_two = MusicMaker(
545     rmaker=rmaker_two,
546     pitches=tenor_3_random_walk_notes,
547     continuous=True,
548     attachment_handler=attachment_handler_one,
549 )
550 tenor_three_musicmaker_three = MusicMaker(
551     rmaker=rmaker_two,
552     pitches=tenor_3_random_walk_notes,
553     continuous=True,
554     attachment_handler=attachment_handler_two,
555 )
556 #####tenor_four#####
557 tenor_four_musicmaker_one = MusicMaker(
558     rmaker=rmaker_one,
559     pitches=tenor_4_chord,
560     continuous=True,
561     attachment_handler=attachment_handler_one,
562 )
563 tenor_four_musicmaker_two = MusicMaker(
564     rmaker=rmaker_two,
565     pitches=tenor_4_random_walk_notes,
566     continuous=True,
567     attachment_handler=attachment_handler_one,
568 )
569 tenor_four_musicmaker_three = MusicMaker(
570     rmaker=rmaker_two,
571     pitches=tenor_4_random_walk_notes,
572     continuous=True,
573     attachment_handler=attachment_handler_two,
574 )
575 #####tenor_five#####
576 tenor_five_musicmaker_one = MusicMaker(
577     rmaker=rmaker_one,
578     pitches=tenor_5_chord,
579     continuous=True,
580     attachment_handler=attachment_handler_one,
581 )
582 tenor_five_musicmaker_two = MusicMaker(
583     rmaker=rmaker_two,
584     pitches=tenor_5_random_walk_notes,
585     continuous=True,
586     attachment_handler=attachment_handler_one,
587 )
588 tenor_five_musicmaker_three = MusicMaker(
```

```
589     rmaker=rmaker_two,
590     pitches=tenor_5_random_walk_notes,
591     continuous=True,
592     attachment_handler=attachment_handler_two,
593 )
594 #####baritone_one#####
595 baritone_one_musicmaker_one = MusicMaker(
596     rmaker=rmaker_one,
597     pitches=baritone_1_chord,
598     continuous=True,
599     attachment_handler=attachment_handler_one,
600 )
601 baritone_one_musicmaker_two = MusicMaker(
602     rmaker=rmaker_two,
603     pitches=baritone_1_random_walk_notes,
604     continuous=True,
605     attachment_handler=attachment_handler_one,
606 )
607 baritone_one_musicmaker_three = MusicMaker(
608     rmaker=rmaker_two,
609     pitches=baritone_1_random_walk_notes,
610     continuous=True,
611     attachment_handler=attachment_handler_two,
612 )
613 #####baritone_two#####
614 baritone_two_musicmaker_one = MusicMaker(
615     rmaker=rmaker_one,
616     pitches=baritone_2_chord,
617     continuous=True,
618     attachment_handler=attachment_handler_one,
619 )
620 baritone_two_musicmaker_two = MusicMaker(
621     rmaker=rmaker_two,
622     pitches=baritone_2_random_walk_notes,
623     continuous=True,
624     attachment_handler=attachment_handler_one,
625 )
626 baritone_two_musicmaker_three = MusicMaker(
627     rmaker=rmaker_two,
628     pitches=baritone_2_random_walk_notes,
629     continuous=True,
630     attachment_handler=attachment_handler_two,
631 )
632 #####baritone_three#####
633 baritone_three_musicmaker_one = MusicMaker(
634     rmaker=rmaker_one,
635     pitches=baritone_3_chord,
636     continuous=True,
637     attachment_handler=attachment_handler_one,
638 )
639 baritone_three_musicmaker_two = MusicMaker(
640     rmaker=rmaker_two,
641     pitches=baritone_3_random_walk_notes,
642     continuous=True,
643     attachment_handler=attachment_handler_one,
644 )
645 baritone_three_musicmaker_three = MusicMaker(
646     rmaker=rmaker_two,
647     pitches=baritone_3_random_walk_notes,
```

```
648     continuous=True,
649     attachment_handler=attachment_handler_two,
650 )
651 #####bass_one#####
652 bass_one_musicmaker_one = MusicMaker(
653     rmaker=rmaker_one,
654     pitches=bass_1_chord,
655     continuous=True,
656     attachment_handler=attachment_handler_one,
657 )
658 bass_one_musicmaker_two = MusicMaker(
659     rmaker=rmaker_two,
660     pitches=bass_1_random_walk_notes,
661     continuous=True,
662     attachment_handler=attachment_handler_one,
663 )
664 bass_one_musicmaker_three = MusicMaker(
665     rmaker=rmaker_two,
666     pitches=bass_1_random_walk_notes,
667     continuous=True,
668     attachment_handler=attachment_handler_two,
669 )
670 #####bass_two#####
671 bass_two_musicmaker_one = MusicMaker(
672     rmaker=rmaker_one,
673     pitches=bass_2_chord,
674     continuous=True,
675     attachment_handler=attachment_handler_one,
676 )
677 bass_two_musicmaker_two = MusicMaker(
678     rmaker=rmaker_two,
679     pitches=bass_2_random_walk_notes,
680     continuous=True,
681     attachment_handler=attachment_handler_one,
682 )
683 bass_two_musicmaker_three = MusicMaker(
684     rmaker=rmaker_two,
685     pitches=bass_2_random_walk_notes,
686     continuous=True,
687     attachment_handler=attachment_handler_two,
688 )
689 #####contrabass#####
690 contrabass_musicmaker_one = MusicMaker(
691     rmaker=rmaker_one,
692     pitches=contrabass_chord,
693     continuous=True,
694     attachment_handler=attachment_handler_one,
695 )
696 contrabass_musicmaker_two = MusicMaker(
697     rmaker=rmaker_two,
698     pitches=contrabass_random_walk_notes,
699     continuous=True,
700     attachment_handler=attachment_handler_one,
701 )
702 contrabass_musicmaker_three = MusicMaker(
703     rmaker=rmaker_two,
704     pitches=contrabass_random_walk_notes,
705     continuous=True,
706     attachment_handler=attachment_handler_two,
```

```

707 )
708
709 silence_maker = abjadext.rmakers.NoteRhythmMaker(
710     division_masks=[
711         abjadext.rmakers.SilenceMask(
712             pattern=abjad.index([0], 1),
713         ),
714     ],
715 )
716
717 class MusicSpecifier:
718
719     def __init__(self, music_maker, voice_name):
720         self.music_maker = music_maker
721         self.voice_name = voice_name
722
723     print('Collecting timespans and rmakers ...')
724
725 voice_1_timespan_list = abjad.TimespanList([
726     abjad.AnnotatedTimespan(
727         start_offset=start_offset,
728         stop_offset=stop_offset,
729         annotation=MusicSpecifier(
730             music_maker=music_maker,
731             voice_name='Voice 1',
732         ),
733     )
734     for start_offset, stop_offset, music_maker in [
735         [(0, 8), (2, 8), sopranino_musicmaker_two],
736         [(2, 8), (4, 8), sopranino_musicmaker_three],
737         [(4, 8), (6, 8), sopranino_musicmaker_two],
738         [(6, 8), (8, 8), sopranino_musicmaker_two],
739         [(8, 8), (10, 8), sopranino_musicmaker_three],
740         [(10, 8), (12, 8), sopranino_musicmaker_two],
741         [(12, 8), (14, 8), sopranino_musicmaker_two],
742         [(14, 8), (16, 8), sopranino_musicmaker_two],
743         [(16, 8), (18, 8), sopranino_musicmaker_three],
744         [(18, 8), (20, 8), sopranino_musicmaker_two],
745         [(20, 8), (22, 8), sopranino_musicmaker_two],
746         [(22, 8), (24, 8), sopranino_musicmaker_two],
747         [(24, 8), (26, 8), sopranino_musicmaker_two],
748         [(26, 8), (28, 8), sopranino_musicmaker_three],
749         [(28, 8), (30, 8), sopranino_musicmaker_two],
750         [(30, 8), (32, 8), sopranino_musicmaker_two],
751         [(32, 8), (34, 8), sopranino_musicmaker_two],
752         [(34, 8), (36, 8), sopranino_musicmaker_two],
753         [(36, 8), (38, 8), sopranino_musicmaker_three],
754         [(38, 8), (40, 8), sopranino_musicmaker_three],
755         [(40, 8), (42, 8), sopranino_musicmaker_two],
756         [(42, 8), (44, 8), sopranino_musicmaker_two],
757         [(44, 8), (46, 8), sopranino_musicmaker_two],
758         [(46, 8), (48, 8), sopranino_musicmaker_three],
759         [(48, 8), (50, 8), sopranino_musicmaker_three],
760         [(50, 8), (52, 8), sopranino_musicmaker_three],
761         [(52, 8), (54, 8), sopranino_musicmaker_two],
762         [(54, 8), (56, 8), sopranino_musicmaker_two],
763         [(56, 8), (58, 8), sopranino_musicmaker_three],
764         [(58, 8), (60, 8), sopranino_musicmaker_three],
765         [(60, 8), (62, 8), sopranino_musicmaker_three],

```

```

766     [(62, 8), (64, 8), soprano_musicmaker_two],
767     [(64, 8), (66, 8), soprano_musicmaker_two],
768     [(66, 8), (68, 8), soprano_musicmaker_three],
769     [(68, 8), (70, 8), soprano_musicmaker_three],
770     [(70, 8), (72, 8), soprano_musicmaker_two],
771     [(72, 8), (74, 8), soprano_musicmaker_three],
772     [(74, 8), (76, 8), soprano_musicmaker_three],
773     [(76, 8), (78, 8), soprano_musicmaker_two],
774     [(78, 8), (80, 8), soprano_musicmaker_three],
775     [(80, 8), (82, 8), soprano_musicmaker_three],
776     [(82, 8), (84, 8), soprano_musicmaker_three],
777     [(84, 8), (86, 8), soprano_musicmaker_three],
778     [(86, 8), (88, 8), soprano_musicmaker_two],
779     [(88, 8), (90, 8), soprano_musicmaker_two],
780     [(90, 8), (92, 8), soprano_musicmaker_two],
781     [(92, 8), (94, 8), soprano_musicmaker_two],
782     [(94, 8), (96, 8), soprano_musicmaker_two],
783 ]
784 ])
785
786 voice_2_timespan_list = abjad.TimespanList([
787     abjad.AnnotatedTimespan(
788         start_offset=start_offset,
789         stop_offset=stop_offset,
790         annotation=MusicSpecifier(
791             music_maker=music_maker,
792             voice_name='Voice 2',
793         ),
794     )
795     for start_offset, stop_offset, music_maker in [
796         [(0, 8), (2, 8), soprano_one_musicmaker_two],
797         [(2, 8), (4, 8), soprano_one_musicmaker_two],
798         [(4, 8), (6, 8), soprano_one_musicmaker_two],
799         [(6, 8), (8, 8), soprano_one_musicmaker_two],
800         [(8, 8), (10, 8), soprano_one_musicmaker_two],
801         [(10, 8), (12, 8), soprano_one_musicmaker_two],
802         [(12, 8), (14, 8), soprano_one_musicmaker_two],
803         [(14, 8), (16, 8), soprano_one_musicmaker_three],
804         [(16, 8), (18, 8), soprano_one_musicmaker_three],
805         [(18, 8), (20, 8), soprano_one_musicmaker_three],
806         [(20, 8), (22, 8), soprano_one_musicmaker_three],
807         [(22, 8), (24, 8), soprano_one_musicmaker_three],
808         [(24, 8), (26, 8), soprano_one_musicmaker_three],
809         [(26, 8), (28, 8), soprano_one_musicmaker_three],
810         [(28, 8), (30, 8), soprano_one_musicmaker_two],
811         [(30, 8), (32, 8), soprano_one_musicmaker_two],
812         [(32, 8), (34, 8), soprano_one_musicmaker_two],
813         [(34, 8), (36, 8), soprano_one_musicmaker_two],
814         [(36, 8), (38, 8), soprano_one_musicmaker_two],
815         [(38, 8), (40, 8), soprano_one_musicmaker_three],
816         [(40, 8), (42, 8), soprano_one_musicmaker_three],
817         [(42, 8), (44, 8), soprano_one_musicmaker_three],
818         [(44, 8), (46, 8), soprano_one_musicmaker_three],
819         [(46, 8), (48, 8), soprano_one_musicmaker_three],
820         [(48, 8), (50, 8), soprano_one_musicmaker_two],
821         [(50, 8), (52, 8), soprano_one_musicmaker_two],
822         [(52, 8), (54, 8), soprano_one_musicmaker_two],
823         [(54, 8), (56, 8), soprano_one_musicmaker_three],
824         [(56, 8), (58, 8), soprano_one_musicmaker_three],

```

```

825     [(58, 8), (60, 8), soprano_one_musicmaker_three],
826     [(60, 8), (62, 8), soprano_one_musicmaker_two],
827     [(62, 8), (64, 8), soprano_one_musicmaker_three],
828     [(64, 8), (66, 8), soprano_one_musicmaker_two],
829     [(66, 8), (68, 8), soprano_one_musicmaker_two],
830     [(68, 8), (70, 8), soprano_one_musicmaker_two],
831     [(70, 8), (72, 8), soprano_one_musicmaker_two],
832     [(72, 8), (74, 8), soprano_one_musicmaker_two],
833     [(74, 8), (76, 8), soprano_one_musicmaker_two],
834     [(76, 8), (78, 8), soprano_one_musicmaker_three],
835     [(78, 8), (80, 8), soprano_one_musicmaker_three],
836     [(80, 8), (82, 8), soprano_one_musicmaker_three],
837     [(82, 8), (84, 8), soprano_one_musicmaker_three],
838     [(84, 8), (86, 8), soprano_one_musicmaker_three],
839     [(86, 8), (88, 8), soprano_one_musicmaker_three],
840     [(88, 8), (90, 8), soprano_one_musicmaker_two],
841     [(90, 8), (92, 8), soprano_one_musicmaker_two],
842     [(92, 8), (94, 8), soprano_one_musicmaker_two],
843     [(94, 8), (96, 8), soprano_one_musicmaker_two],
844 ]
845 ])
846
847 voice_3_timestspan_list = abjad.TimespanList([
848     abjad.AnnotatedTimespan(
849         start_offset=start_offset,
850         stop_offset=stop_offset,
851         annotation=MusicSpecifier(
852             music_maker=music_maker,
853             voice_name='Voice 3',
854         ),
855     )
856     for start_offset, stop_offset, music_maker in [
857         [(0, 8), (2, 8), soprano_two_musicmaker_three],
858         [(2, 8), (4, 8), soprano_two_musicmaker_three],
859         [(4, 8), (6, 8), soprano_two_musicmaker_three],
860         [(6, 8), (8, 8), soprano_two_musicmaker_three],
861         [(8, 8), (10, 8), soprano_two_musicmaker_three],
862         [(10, 8), (12, 8), soprano_two_musicmaker_three],
863         [(12, 8), (14, 8), soprano_two_musicmaker_three],
864         [(14, 8), (16, 8), soprano_two_musicmaker_three],
865         [(16, 8), (18, 8), soprano_two_musicmaker_two],
866         [(18, 8), (20, 8), soprano_two_musicmaker_two],
867         [(20, 8), (22, 8), soprano_two_musicmaker_two],
868         [(22, 8), (24, 8), soprano_two_musicmaker_two],
869         [(24, 8), (26, 8), soprano_two_musicmaker_two],
870         [(26, 8), (28, 8), soprano_two_musicmaker_two],
871         [(28, 8), (30, 8), soprano_two_musicmaker_two],
872         [(30, 8), (32, 8), soprano_two_musicmaker_three],
873         [(32, 8), (34, 8), soprano_two_musicmaker_three],
874         [(34, 8), (36, 8), soprano_two_musicmaker_three],
875         [(36, 8), (38, 8), soprano_two_musicmaker_three],
876         [(38, 8), (40, 8), soprano_two_musicmaker_three],
877         [(40, 8), (42, 8), soprano_two_musicmaker_three],
878         [(42, 8), (44, 8), soprano_two_musicmaker_two],
879         [(44, 8), (46, 8), soprano_two_musicmaker_two],
880         [(46, 8), (48, 8), soprano_two_musicmaker_two],
881         [(48, 8), (50, 8), soprano_two_musicmaker_two],
882         [(50, 8), (52, 8), soprano_two_musicmaker_three],
883         [(52, 8), (54, 8), soprano_two_musicmaker_three],

```

```

884     [(54, 8), (56, 8), soprano_two_musicmaker_three],
885     [(56, 8), (58, 8), soprano_two_musicmaker_three],
886     [(58, 8), (60, 8), soprano_two_musicmaker_two],
887     [(60, 8), (62, 8), soprano_two_musicmaker_two],
888     [(62, 8), (64, 8), soprano_two_musicmaker_two],
889     [(64, 8), (66, 8), soprano_two_musicmaker_three],
890     [(66, 8), (68, 8), soprano_two_musicmaker_three],
891     [(68, 8), (70, 8), soprano_two_musicmaker_two],
892     [(70, 8), (72, 8), soprano_two_musicmaker_three],
893     [(72, 8), (74, 8), soprano_two_musicmaker_two],
894     [(74, 8), (76, 8), soprano_two_musicmaker_two],
895     [(76, 8), (78, 8), soprano_two_musicmaker_three],
896     [(78, 8), (80, 8), soprano_two_musicmaker_three],
897     [(80, 8), (82, 8), soprano_two_musicmaker_three],
898     [(82, 8), (84, 8), soprano_two_musicmaker_two],
899     [(84, 8), (86, 8), soprano_two_musicmaker_two],
900     [(86, 8), (88, 8), soprano_two_musicmaker_two],
901     [(88, 8), (90, 8), soprano_two_musicmaker_two],
902     [(90, 8), (92, 8), soprano_two_musicmaker_three],
903     [(92, 8), (94, 8), soprano_two_musicmaker_three],
904     [(94, 8), (96, 8), soprano_two_musicmaker_three],
905   ],
906 ]
907
908 voice_4_timespan_list = abjad.TimespanList([
909     abjad.AnnotatedTimespan(
910         start_offset=start_offset,
911         stop_offset=stop_offset,
912         annotation=MusicSpecifier(
913             music_maker=music_maker,
914             voice_name='Voice 4',
915         ),
916     ),
917     for start_offset, stop_offset, music_maker in [
918         [(0, 8), (2, 8), soprano_three_musicmaker_two],
919         [(2, 8), (4, 8), soprano_three_musicmaker_two],
920         [(4, 8), (6, 8), soprano_three_musicmaker_two],
921         [(6, 8), (8, 8), soprano_three_musicmaker_two],
922         [(8, 8), (10, 8), soprano_three_musicmaker_three],
923         [(10, 8), (12, 8), soprano_three_musicmaker_three],
924         [(12, 8), (14, 8), soprano_three_musicmaker_three],
925         [(14, 8), (16, 8), soprano_three_musicmaker_three],
926         [(16, 8), (18, 8), soprano_three_musicmaker_two],
927         [(18, 8), (20, 8), soprano_three_musicmaker_two],
928         [(20, 8), (22, 8), soprano_three_musicmaker_two],
929         [(22, 8), (24, 8), soprano_three_musicmaker_two],
930         [(24, 8), (26, 8), soprano_three_musicmaker_two],
931         [(26, 8), (28, 8), soprano_three_musicmaker_two],
932         [(28, 8), (30, 8), soprano_three_musicmaker_two],
933         [(30, 8), (32, 8), soprano_three_musicmaker_three],
934         [(32, 8), (34, 8), soprano_three_musicmaker_three],
935         [(34, 8), (36, 8), soprano_three_musicmaker_three],
936         [(36, 8), (38, 8), soprano_three_musicmaker_three],
937         [(38, 8), (40, 8), soprano_three_musicmaker_three],
938         [(40, 8), (42, 8), soprano_three_musicmaker_three],
939         [(42, 8), (44, 8), soprano_three_musicmaker_three],
940         [(44, 8), (46, 8), soprano_three_musicmaker_two],
941         [(46, 8), (48, 8), soprano_three_musicmaker_two],
942         [(48, 8), (50, 8), soprano_three_musicmaker_two],

```

```

943     [(50, 8), (52, 8), soprano_three_musicmaker_two],
944     [(52, 8), (54, 8), soprano_three_musicmaker_two],
945     [(54, 8), (56, 8), soprano_three_musicmaker_two],
946     [(56, 8), (58, 8), soprano_three_musicmaker_two],
947     [(58, 8), (60, 8), soprano_three_musicmaker_three],
948     [(60, 8), (62, 8), soprano_three_musicmaker_three],
949     [(62, 8), (64, 8), soprano_three_musicmaker_three],
950     [(64, 8), (66, 8), soprano_three_musicmaker_three],
951     [(66, 8), (68, 8), soprano_three_musicmaker_three],
952     [(68, 8), (70, 8), soprano_three_musicmaker_three],
953     [(70, 8), (72, 8), soprano_three_musicmaker_three],
954     [(72, 8), (74, 8), soprano_three_musicmaker_two],
955     [(74, 8), (76, 8), soprano_three_musicmaker_two],
956     [(76, 8), (78, 8), soprano_three_musicmaker_two],
957     [(78, 8), (80, 8), soprano_three_musicmaker_two],
958     [(80, 8), (82, 8), soprano_three_musicmaker_two],
959     [(82, 8), (84, 8), soprano_three_musicmaker_two],
960     [(84, 8), (86, 8), soprano_three_musicmaker_two],
961     [(86, 8), (88, 8), soprano_three_musicmaker_three],
962     [(88, 8), (90, 8), soprano_three_musicmaker_three],
963     [(90, 8), (92, 8), soprano_three_musicmaker_three],
964     [(92, 8), (94, 8), soprano_three_musicmaker_three],
965     [(94, 8), (96, 8), soprano_three_musicmaker_three],
966   ]
967 ])
968
969 voice_5_timespan_list = abjad.TimespanList([
970     abjad.AnnotatedTimespan(
971         start_offset=start_offset,
972         stop_offset=stop_offset,
973         annotation=MusicSpecifier(
974             music_maker=music_maker,
975             voice_name='Voice 5',
976         ),
977     )
978     for start_offset, stop_offset, music_maker in [
979         [(0, 8), (2, 8), alto_one_musicmaker_three],
980         [(2, 8), (4, 8), alto_one_musicmaker_three],
981         [(4, 8), (6, 8), alto_one_musicmaker_two],
982         [(6, 8), (8, 8), alto_one_musicmaker_two],
983         [(8, 8), (10, 8), alto_one_musicmaker_two],
984         [(10, 8), (12, 8), alto_one_musicmaker_two],
985         [(12, 8), (14, 8), alto_one_musicmaker_two],
986         [(14, 8), (16, 8), alto_one_musicmaker_three],
987         [(16, 8), (18, 8), alto_one_musicmaker_three],
988         [(18, 8), (20, 8), alto_one_musicmaker_three],
989         [(20, 8), (22, 8), alto_one_musicmaker_three],
990         [(22, 8), (24, 8), alto_one_musicmaker_three],
991         [(24, 8), (26, 8), alto_one_musicmaker_two],
992         [(26, 8), (28, 8), alto_one_musicmaker_two],
993         [(28, 8), (30, 8), alto_one_musicmaker_two],
994         [(30, 8), (32, 8), alto_one_musicmaker_two],
995         [(32, 8), (34, 8), alto_one_musicmaker_two],
996         [(34, 8), (36, 8), alto_one_musicmaker_three],
997         [(36, 8), (38, 8), alto_one_musicmaker_three],
998         [(38, 8), (40, 8), alto_one_musicmaker_three],
999         [(40, 8), (42, 8), alto_one_musicmaker_three],
1000        [(42, 8), (44, 8), alto_one_musicmaker_three],
1001        [(44, 8), (46, 8), alto_one_musicmaker_two],

```

```

1002     [(46, 8), (48, 8), alto_one_musicmaker_two],
1003     [(48, 8), (50, 8), alto_one_musicmaker_two],
1004     [(50, 8), (52, 8), alto_one_musicmaker_two],
1005     [(52, 8), (54, 8), alto_one_musicmaker_two],
1006     [(54, 8), (56, 8), alto_one_musicmaker_three],
1007     [(56, 8), (58, 8), alto_one_musicmaker_three],
1008     [(58, 8), (60, 8), alto_one_musicmaker_three],
1009     [(60, 8), (62, 8), alto_one_musicmaker_three],
1010     [(62, 8), (64, 8), alto_one_musicmaker_three],
1011     [(64, 8), (66, 8), alto_one_musicmaker_two],
1012     [(66, 8), (68, 8), alto_one_musicmaker_two],
1013     [(68, 8), (70, 8), alto_one_musicmaker_two],
1014     [(70, 8), (72, 8), alto_one_musicmaker_two],
1015     [(72, 8), (74, 8), alto_one_musicmaker_two],
1016     [(74, 8), (76, 8), alto_one_musicmaker_three],
1017     [(76, 8), (78, 8), alto_one_musicmaker_three],
1018     [(78, 8), (80, 8), alto_one_musicmaker_three],
1019     [(80, 8), (82, 8), alto_one_musicmaker_three],
1020     [(82, 8), (84, 8), alto_one_musicmaker_three],
1021     [(84, 8), (86, 8), alto_one_musicmaker_two],
1022     [(86, 8), (88, 8), alto_one_musicmaker_two],
1023     [(88, 8), (90, 8), alto_one_musicmaker_two],
1024     [(90, 8), (92, 8), alto_one_musicmaker_two],
1025     [(92, 8), (94, 8), alto_one_musicmaker_two],
1026     [(94, 8), (96, 8), alto_one_musicmaker_three],
1027 ]
1028 ])
1029
1030 voice_6_timespan_list = abjad.TimespanList([
1031     abjad.AnnotatedTimespan(
1032         start_offset=start_offset,
1033         stop_offset=stop_offset,
1034         annotation=MusicSpecifier(
1035             music_maker=music_maker,
1036             voice_name='Voice 6',
1037         ),
1038     ),
1039     for start_offset, stop_offset, music_maker in [
1040         [(0, 8), (2, 8), alto_two_musicmaker_three],
1041         [(2, 8), (4, 8), alto_two_musicmaker_three],
1042         [(4, 8), (6, 8), alto_two_musicmaker_three],
1043         [(6, 8), (8, 8), alto_two_musicmaker_three],
1044         [(8, 8), (10, 8), alto_two_musicmaker_two],
1045         [(10, 8), (12, 8), alto_two_musicmaker_two],
1046         [(12, 8), (14, 8), alto_two_musicmaker_two],
1047         [(14, 8), (16, 8), alto_two_musicmaker_two],
1048         [(16, 8), (18, 8), alto_two_musicmaker_two],
1049         [(18, 8), (20, 8), alto_two_musicmaker_two],
1050         [(20, 8), (22, 8), alto_two_musicmaker_three],
1051         [(22, 8), (24, 8), alto_two_musicmaker_three],
1052         [(24, 8), (26, 8), alto_two_musicmaker_three],
1053         [(26, 8), (28, 8), alto_two_musicmaker_three],
1054         [(28, 8), (30, 8), alto_two_musicmaker_three],
1055         [(30, 8), (32, 8), alto_two_musicmaker_two],
1056         [(32, 8), (34, 8), alto_two_musicmaker_two],
1057         [(34, 8), (36, 8), alto_two_musicmaker_two],
1058         [(36, 8), (38, 8), alto_two_musicmaker_two],
1059         [(38, 8), (40, 8), alto_two_musicmaker_three],
1060         [(40, 8), (42, 8), alto_two_musicmaker_three],

```

```

1061     [(42, 8), (44, 8), alto_two_musicmaker_three],
1062     [(44, 8), (46, 8), alto_two_musicmaker_two],
1063     [(46, 8), (48, 8), alto_two_musicmaker_two],
1064     [(48, 8), (50, 8), alto_two_musicmaker_three],
1065     [(50, 8), (52, 8), alto_two_musicmaker_two],
1066     [(52, 8), (54, 8), alto_two_musicmaker_three],
1067     [(54, 8), (56, 8), alto_two_musicmaker_two],
1068     [(56, 8), (58, 8), alto_two_musicmaker_two],
1069     [(58, 8), (60, 8), alto_two_musicmaker_three],
1070     [(60, 8), (62, 8), alto_two_musicmaker_three],
1071     [(62, 8), (64, 8), alto_two_musicmaker_two],
1072     [(64, 8), (66, 8), alto_two_musicmaker_two],
1073     [(66, 8), (68, 8), alto_two_musicmaker_two],
1074     [(68, 8), (70, 8), alto_two_musicmaker_three],
1075     [(70, 8), (72, 8), alto_two_musicmaker_three],
1076     [(72, 8), (74, 8), alto_two_musicmaker_three],
1077     [(74, 8), (76, 8), alto_two_musicmaker_two],
1078     [(76, 8), (78, 8), alto_two_musicmaker_two],
1079     [(78, 8), (80, 8), alto_two_musicmaker_two],
1080     [(80, 8), (82, 8), alto_two_musicmaker_two],
1081     [(82, 8), (84, 8), alto_two_musicmaker_three],
1082     [(84, 8), (86, 8), alto_two_musicmaker_three],
1083     [(86, 8), (88, 8), alto_two_musicmaker_three],
1084     [(88, 8), (90, 8), alto_two_musicmaker_three],
1085     [(90, 8), (92, 8), alto_two_musicmaker_two],
1086     [(92, 8), (94, 8), alto_two_musicmaker_two],
1087     [(94, 8), (96, 8), alto_two_musicmaker_two],
1088 ]
1089 ])
1090
1091 voice_7_timespan_list = abjad.TimespanList([
1092     abjad.AnnotatedTimespan(
1093         start_offset=start_offset,
1094         stop_offset=stop_offset,
1095         annotation=MusicSpecifier(
1096             music_maker=music_maker,
1097             voice_name='Voice 7',
1098         ),
1099     )
1100     for start_offset, stop_offset, music_maker in [
1101         [(0, 8), (2, 8), alto_three_musicmaker_two],
1102         [(2, 8), (4, 8), alto_three_musicmaker_two],
1103         [(4, 8), (6, 8), alto_three_musicmaker_two],
1104         [(6, 8), (8, 8), alto_three_musicmaker_two],
1105         [(8, 8), (10, 8), alto_three_musicmaker_two],
1106         [(10, 8), (12, 8), alto_three_musicmaker_three],
1107         [(12, 8), (14, 8), alto_three_musicmaker_two],
1108         [(14, 8), (16, 8), alto_three_musicmaker_three],
1109         [(16, 8), (18, 8), alto_three_musicmaker_three],
1110         [(18, 8), (20, 8), alto_three_musicmaker_three],
1111         [(20, 8), (22, 8), alto_three_musicmaker_two],
1112         [(22, 8), (24, 8), alto_three_musicmaker_two],
1113         [(24, 8), (26, 8), alto_three_musicmaker_two],
1114         [(26, 8), (28, 8), alto_three_musicmaker_two],
1115         [(28, 8), (30, 8), alto_three_musicmaker_two],
1116         [(30, 8), (32, 8), alto_three_musicmaker_three],
1117         [(32, 8), (34, 8), alto_three_musicmaker_three],
1118         [(34, 8), (36, 8), alto_three_musicmaker_three],
1119         [(36, 8), (38, 8), alto_three_musicmaker_three],

```

```

1120     [(38, 8), (40, 8), alto_three_musicmaker_three],
1121     [(40, 8), (42, 8), alto_three_musicmaker_three],
1122     [(42, 8), (44, 8), alto_three_musicmaker_two],
1123     [(44, 8), (46, 8), alto_three_musicmaker_two],
1124     [(46, 8), (48, 8), alto_three_musicmaker_two],
1125     [(48, 8), (50, 8), alto_three_musicmaker_two],
1126     [(50, 8), (52, 8), alto_three_musicmaker_two],
1127     [(52, 8), (54, 8), alto_three_musicmaker_two],
1128     [(54, 8), (56, 8), alto_three_musicmaker_two],
1129     [(56, 8), (58, 8), alto_three_musicmaker_three],
1130     [(58, 8), (60, 8), alto_three_musicmaker_three],
1131     [(60, 8), (62, 8), alto_three_musicmaker_three],
1132     [(62, 8), (64, 8), alto_three_musicmaker_three],
1133     [(64, 8), (66, 8), alto_three_musicmaker_two],
1134     [(66, 8), (68, 8), alto_three_musicmaker_two],
1135     [(68, 8), (70, 8), alto_three_musicmaker_two],
1136     [(70, 8), (72, 8), alto_three_musicmaker_two],
1137     [(72, 8), (74, 8), alto_three_musicmaker_two],
1138     [(74, 8), (76, 8), alto_three_musicmaker_three],
1139     [(76, 8), (78, 8), alto_three_musicmaker_three],
1140     [(78, 8), (80, 8), alto_three_musicmaker_three],
1141     [(80, 8), (82, 8), alto_three_musicmaker_three],
1142     [(82, 8), (84, 8), alto_three_musicmaker_three],
1143     [(84, 8), (86, 8), alto_three_musicmaker_three],
1144     [(86, 8), (88, 8), alto_three_musicmaker_two],
1145     [(88, 8), (90, 8), alto_three_musicmaker_two],
1146     [(90, 8), (92, 8), alto_three_musicmaker_two],
1147     [(92, 8), (94, 8), alto_three_musicmaker_two],
1148     [(94, 8), (96, 8), alto_three_musicmaker_two],
1149 ]
1150 ])
1151
1152 voice_8_timespan_list = abjad.TimespanList([
1153     abjad.AnnotatedTimespan(
1154         start_offset=start_offset,
1155         stop_offset=stop_offset,
1156         annotation=MusicSpecifier(
1157             music_maker=music_maker,
1158             voice_name='Voice 8',
1159         ),
1160     )
1161     for start_offset, stop_offset, music_maker in [
1162         [(0, 8), (2, 8), alto_four_musicmaker_two],
1163         [(2, 8), (4, 8), alto_four_musicmaker_two],
1164         [(4, 8), (6, 8), alto_four_musicmaker_two],
1165         [(6, 8), (8, 8), alto_four_musicmaker_three],
1166         [(8, 8), (10, 8), alto_four_musicmaker_three],
1167         [(10, 8), (12, 8), alto_four_musicmaker_three],
1168         [(12, 8), (14, 8), alto_four_musicmaker_two],
1169         [(14, 8), (16, 8), alto_four_musicmaker_two],
1170         [(16, 8), (18, 8), alto_four_musicmaker_two],
1171         [(18, 8), (20, 8), alto_four_musicmaker_three],
1172         [(20, 8), (22, 8), alto_four_musicmaker_three],
1173         [(22, 8), (24, 8), alto_four_musicmaker_three],
1174         [(24, 8), (26, 8), alto_four_musicmaker_three],
1175         [(26, 8), (28, 8), alto_four_musicmaker_two],
1176         [(28, 8), (30, 8), alto_four_musicmaker_two],
1177         [(30, 8), (32, 8), alto_four_musicmaker_two],
1178         [(32, 8), (34, 8), alto_four_musicmaker_two],

```

```

1179     [(34, 8), (36, 8), alto_four_musicmaker_three],
1180     [(36, 8), (38, 8), alto_four_musicmaker_three],
1181     [(38, 8), (40, 8), alto_four_musicmaker_three],
1182     [(40, 8), (42, 8), alto_four_musicmaker_three],
1183     [(42, 8), (44, 8), alto_four_musicmaker_three],
1184     [(44, 8), (46, 8), alto_four_musicmaker_two],
1185     [(46, 8), (48, 8), alto_four_musicmaker_two],
1186     [(48, 8), (50, 8), alto_four_musicmaker_two],
1187     [(50, 8), (52, 8), alto_four_musicmaker_two],
1188     [(52, 8), (54, 8), alto_four_musicmaker_two],
1189     [(54, 8), (56, 8), alto_four_musicmaker_three],
1190     [(56, 8), (58, 8), alto_four_musicmaker_three],
1191     [(58, 8), (60, 8), alto_four_musicmaker_three],
1192     [(60, 8), (62, 8), alto_four_musicmaker_three],
1193     [(62, 8), (64, 8), alto_four_musicmaker_three],
1194     [(64, 8), (66, 8), alto_four_musicmaker_three],
1195     [(66, 8), (68, 8), alto_four_musicmaker_two],
1196     [(68, 8), (70, 8), alto_four_musicmaker_two],
1197     [(70, 8), (72, 8), alto_four_musicmaker_two],
1198     [(72, 8), (74, 8), alto_four_musicmaker_two],
1199     [(74, 8), (76, 8), alto_four_musicmaker_two],
1200     [(76, 8), (78, 8), alto_four_musicmaker_two],
1201     [(78, 8), (80, 8), alto_four_musicmaker_three],
1202     [(80, 8), (82, 8), alto_four_musicmaker_three],
1203     [(82, 8), (84, 8), alto_four_musicmaker_three],
1204     [(84, 8), (86, 8), alto_four_musicmaker_three],
1205     [(86, 8), (88, 8), alto_four_musicmaker_three],
1206     [(88, 8), (90, 8), alto_four_musicmaker_three],
1207     [(90, 8), (92, 8), alto_four_musicmaker_three],
1208     [(92, 8), (94, 8), alto_four_musicmaker_two],
1209     [(94, 8), (96, 8), alto_four_musicmaker_two],
1210   ],
1211 ])
1212
1213 voice_9_timespan_list = abjad.TimespanList([
1214     abjad.AnnotatedTimespan(
1215         start_offset=start_offset,
1216         stop_offset=stop_offset,
1217         annotation=MusicSpecifier(
1218             music_maker=music_maker,
1219             voice_name='Voice 9',
1220         ),
1221     )
1222     for start_offset, stop_offset, music_maker in [
1223         [(0, 8), (2, 8), alto_five_musicmaker_two],
1224         [(2, 8), (4, 8), alto_five_musicmaker_three],
1225         [(4, 8), (6, 8), alto_five_musicmaker_three],
1226         [(6, 8), (8, 8), alto_five_musicmaker_two],
1227         [(8, 8), (10, 8), alto_five_musicmaker_two],
1228         [(10, 8), (12, 8), alto_five_musicmaker_two],
1229         [(12, 8), (14, 8), alto_five_musicmaker_three],
1230         [(14, 8), (16, 8), alto_five_musicmaker_three],
1231         [(16, 8), (18, 8), alto_five_musicmaker_three],
1232         [(18, 8), (20, 8), alto_five_musicmaker_three],
1233         [(20, 8), (22, 8), alto_five_musicmaker_two],
1234         [(22, 8), (24, 8), alto_five_musicmaker_two],
1235         [(24, 8), (26, 8), alto_five_musicmaker_two],
1236         [(26, 8), (28, 8), alto_five_musicmaker_two],
1237         [(28, 8), (30, 8), alto_five_musicmaker_two],

```

```

1238     [(30, 8), (32, 8), alto_five_musicmaker_three],
1239     [(32, 8), (34, 8), alto_five_musicmaker_three],
1240     [(34, 8), (36, 8), alto_five_musicmaker_three],
1241     [(36, 8), (38, 8), alto_five_musicmaker_three],
1242     [(38, 8), (40, 8), alto_five_musicmaker_three],
1243     [(40, 8), (42, 8), alto_five_musicmaker_three],
1244     [(42, 8), (44, 8), alto_five_musicmaker_two],
1245     [(44, 8), (46, 8), alto_five_musicmaker_two],
1246     [(46, 8), (48, 8), alto_five_musicmaker_two],
1247     [(48, 8), (50, 8), alto_five_musicmaker_two],
1248     [(50, 8), (52, 8), alto_five_musicmaker_two],
1249     [(52, 8), (54, 8), alto_five_musicmaker_two],
1250     [(54, 8), (56, 8), alto_five_musicmaker_two],
1251     [(56, 8), (58, 8), alto_five_musicmaker_three],
1252     [(58, 8), (60, 8), alto_five_musicmaker_three],
1253     [(60, 8), (62, 8), alto_five_musicmaker_three],
1254     [(62, 8), (64, 8), alto_five_musicmaker_three],
1255     [(64, 8), (66, 8), alto_five_musicmaker_three],
1256     [(66, 8), (68, 8), alto_five_musicmaker_three],
1257     [(68, 8), (70, 8), alto_five_musicmaker_three],
1258     [(70, 8), (72, 8), alto_five_musicmaker_three],
1259     [(72, 8), (74, 8), alto_five_musicmaker_two],
1260     [(74, 8), (76, 8), alto_five_musicmaker_two],
1261     [(76, 8), (78, 8), alto_five_musicmaker_two],
1262     [(78, 8), (80, 8), alto_five_musicmaker_two],
1263     [(80, 8), (82, 8), alto_five_musicmaker_two],
1264     [(82, 8), (84, 8), alto_five_musicmaker_two],
1265     [(84, 8), (86, 8), alto_five_musicmaker_two],
1266     [(86, 8), (88, 8), alto_five_musicmaker_two],
1267     [(88, 8), (90, 8), alto_five_musicmaker_three],
1268     [(90, 8), (92, 8), alto_five_musicmaker_three],
1269     [(92, 8), (94, 8), alto_five_musicmaker_three],
1270     [(94, 8), (96, 8), alto_five_musicmaker_three],
1271 ]
1272 ])
1273
1274 voice_10_timespan_list = abjad.TimespanList([
1275     abjad.AnnotatedTimespan(
1276         start_offset=start_offset,
1277         stop_offset=stop_offset,
1278         annotation=MusicSpecifier(
1279             music_maker=music_maker,
1280             voice_name='Voice 10',
1281         ),
1282     )
1283     for start_offset, stop_offset, music_maker in [
1284         [(0, 8), (2, 8), alto_six_musicmaker_three],
1285         [(2, 8), (4, 8), alto_six_musicmaker_three],
1286         [(4, 8), (6, 8), alto_six_musicmaker_three],
1287         [(6, 8), (8, 8), alto_six_musicmaker_three],
1288         [(8, 8), (10, 8), alto_six_musicmaker_two],
1289         [(10, 8), (12, 8), alto_six_musicmaker_two],
1290         [(12, 8), (14, 8), alto_six_musicmaker_two],
1291         [(14, 8), (16, 8), alto_six_musicmaker_two],
1292         [(16, 8), (18, 8), alto_six_musicmaker_three],
1293         [(18, 8), (20, 8), alto_six_musicmaker_three],
1294         [(20, 8), (22, 8), alto_six_musicmaker_three],
1295         [(22, 8), (24, 8), alto_six_musicmaker_three],
1296         [(24, 8), (26, 8), alto_six_musicmaker_three],

```

```

1297     [(26, 8), (28, 8), alto_six_musicmaker_two],
1298     [(28, 8), (30, 8), alto_six_musicmaker_two],
1299     [(30, 8), (32, 8), alto_six_musicmaker_two],
1300     [(32, 8), (34, 8), alto_six_musicmaker_two],
1301     [(34, 8), (36, 8), alto_six_musicmaker_two],
1302     [(36, 8), (38, 8), alto_six_musicmaker_two],
1303     [(38, 8), (40, 8), alto_six_musicmaker_three],
1304     [(40, 8), (42, 8), alto_six_musicmaker_three],
1305     [(42, 8), (44, 8), alto_six_musicmaker_three],
1306     [(44, 8), (46, 8), alto_six_musicmaker_three],
1307     [(46, 8), (48, 8), alto_six_musicmaker_three],
1308     [(48, 8), (50, 8), alto_six_musicmaker_three],
1309     [(50, 8), (52, 8), alto_six_musicmaker_three],
1310     [(52, 8), (54, 8), alto_six_musicmaker_two],
1311     [(54, 8), (56, 8), alto_six_musicmaker_two],
1312     [(56, 8), (58, 8), alto_six_musicmaker_two],
1313     [(58, 8), (60, 8), alto_six_musicmaker_two],
1314     [(60, 8), (62, 8), alto_six_musicmaker_two],
1315     [(62, 8), (64, 8), alto_six_musicmaker_two],
1316     [(64, 8), (66, 8), alto_six_musicmaker_two],
1317     [(66, 8), (68, 8), alto_six_musicmaker_three],
1318     [(68, 8), (70, 8), alto_six_musicmaker_three],
1319     [(70, 8), (72, 8), alto_six_musicmaker_three],
1320     [(72, 8), (74, 8), alto_six_musicmaker_three],
1321     [(74, 8), (76, 8), alto_six_musicmaker_three],
1322     [(76, 8), (78, 8), alto_six_musicmaker_three],
1323     [(78, 8), (80, 8), alto_six_musicmaker_two],
1324     [(80, 8), (82, 8), alto_six_musicmaker_two],
1325     [(82, 8), (84, 8), alto_six_musicmaker_two],
1326     [(84, 8), (86, 8), alto_six_musicmaker_two],
1327     [(86, 8), (88, 8), alto_six_musicmaker_two],
1328     [(88, 8), (90, 8), alto_six_musicmaker_three],
1329     [(90, 8), (92, 8), alto_six_musicmaker_three],
1330     [(92, 8), (94, 8), alto_six_musicmaker_three],
1331     [(94, 8), (96, 8), alto_six_musicmaker_three],
1332   ],
1333 ]
1334
1335 voice_11_timespan_list = abjad.TimespanList([
1336     abjad.AnnotatedTimespan(
1337         start_offset=start_offset,
1338         stop_offset=stop_offset,
1339         annotation=MusicSpecifier(
1340             music_maker=music_maker,
1341             voice_name='Voice 11',
1342         ),
1343     )
1344     for start_offset, stop_offset, music_maker in [
1345         [(0, 8), (2, 8), tenor_one_musicmaker_three],
1346         [(2, 8), (4, 8), tenor_one_musicmaker_three],
1347         [(4, 8), (6, 8), tenor_one_musicmaker_two],
1348         [(6, 8), (8, 8), tenor_one_musicmaker_two],
1349         [(8, 8), (10, 8), tenor_one_musicmaker_two],
1350         [(10, 8), (12, 8), tenor_one_musicmaker_three],
1351         [(12, 8), (14, 8), tenor_one_musicmaker_three],
1352         [(14, 8), (16, 8), tenor_one_musicmaker_three],
1353         [(16, 8), (18, 8), tenor_one_musicmaker_three],
1354         [(18, 8), (20, 8), tenor_one_musicmaker_two],
1355         [(20, 8), (22, 8), tenor_one_musicmaker_two],

```

```

1356     [(22, 8), (24, 8), tenor_one_musicmaker_two],
1357     [(24, 8), (26, 8), tenor_one_musicmaker_two],
1358     [(26, 8), (28, 8), tenor_one_musicmaker_three],
1359     [(28, 8), (30, 8), tenor_one_musicmaker_three],
1360     [(30, 8), (32, 8), tenor_one_musicmaker_three],
1361     [(32, 8), (34, 8), tenor_one_musicmaker_three],
1362     [(34, 8), (36, 8), tenor_one_musicmaker_three],
1363     [(36, 8), (38, 8), tenor_one_musicmaker_two],
1364     [(38, 8), (40, 8), tenor_one_musicmaker_two],
1365     [(40, 8), (42, 8), tenor_one_musicmaker_two],
1366     [(42, 8), (44, 8), tenor_one_musicmaker_two],
1367     [(44, 8), (46, 8), tenor_one_musicmaker_three],
1368     [(46, 8), (48, 8), tenor_one_musicmaker_three],
1369     [(48, 8), (50, 8), tenor_one_musicmaker_three],
1370     [(50, 8), (52, 8), tenor_one_musicmaker_three],
1371     [(52, 8), (54, 8), tenor_one_musicmaker_two],
1372     [(54, 8), (56, 8), tenor_one_musicmaker_two],
1373     [(56, 8), (58, 8), tenor_one_musicmaker_two],
1374     [(58, 8), (60, 8), tenor_one_musicmaker_three],
1375     [(60, 8), (62, 8), tenor_one_musicmaker_three],
1376     [(62, 8), (64, 8), tenor_one_musicmaker_three],
1377     [(64, 8), (66, 8), tenor_one_musicmaker_two],
1378     [(66, 8), (68, 8), tenor_one_musicmaker_two],
1379     [(68, 8), (70, 8), tenor_one_musicmaker_three],
1380     [(70, 8), (72, 8), tenor_one_musicmaker_two],
1381     [(72, 8), (74, 8), tenor_one_musicmaker_two],
1382     [(74, 8), (76, 8), tenor_one_musicmaker_two],
1383     [(76, 8), (78, 8), tenor_one_musicmaker_three],
1384     [(78, 8), (80, 8), tenor_one_musicmaker_three],
1385     [(80, 8), (82, 8), tenor_one_musicmaker_three],
1386     [(82, 8), (84, 8), tenor_one_musicmaker_three],
1387     [(84, 8), (86, 8), tenor_one_musicmaker_two],
1388     [(86, 8), (88, 8), tenor_one_musicmaker_two],
1389     [(88, 8), (90, 8), tenor_one_musicmaker_two],
1390     [(90, 8), (92, 8), tenor_one_musicmaker_two],
1391     [(92, 8), (94, 8), tenor_one_musicmaker_two],
1392     [(94, 8), (96, 8), tenor_one_musicmaker_two],
1393 ]
1394 ])
1395
1396 voice_12_timespan_list = abjad.TimespanList([
1397     abjad.AnnotatedTimespan(
1398         start_offset=start_offset,
1399         stop_offset=stop_offset,
1400         annotation=MusicSpecifier(
1401             music_maker=music_maker,
1402             voice_name='Voice 12',
1403         ),
1404     )
1405     for start_offset, stop_offset, music_maker in [
1406         [(0, 8), (2, 8), tenor_two_musicmaker_two],
1407         [(2, 8), (4, 8), tenor_two_musicmaker_three],
1408         [(4, 8), (6, 8), tenor_two_musicmaker_three],
1409         [(6, 8), (8, 8), tenor_two_musicmaker_three],
1410         [(8, 8), (10, 8), tenor_two_musicmaker_three],
1411         [(10, 8), (12, 8), tenor_two_musicmaker_three],
1412         [(12, 8), (14, 8), tenor_two_musicmaker_three],
1413         [(14, 8), (16, 8), tenor_two_musicmaker_two],
1414         [(16, 8), (18, 8), tenor_two_musicmaker_two],

```

```

1415     [(18, 8), (20, 8), tenor_two_musicmaker_two],
1416     [(20, 8), (22, 8), tenor_two_musicmaker_two],
1417     [(22, 8), (24, 8), tenor_two_musicmaker_two],
1418     [(24, 8), (26, 8), tenor_two_musicmaker_two],
1419     [(26, 8), (28, 8), tenor_two_musicmaker_two],
1420     [(28, 8), (30, 8), tenor_two_musicmaker_three],
1421     [(30, 8), (32, 8), tenor_two_musicmaker_three],
1422     [(32, 8), (34, 8), tenor_two_musicmaker_three],
1423     [(34, 8), (36, 8), tenor_two_musicmaker_three],
1424     [(36, 8), (38, 8), tenor_two_musicmaker_three],
1425     [(38, 8), (40, 8), tenor_two_musicmaker_three],
1426     [(40, 8), (42, 8), tenor_two_musicmaker_three],
1427     [(42, 8), (44, 8), tenor_two_musicmaker_two],
1428     [(44, 8), (46, 8), tenor_two_musicmaker_two],
1429     [(46, 8), (48, 8), tenor_two_musicmaker_two],
1430     [(48, 8), (50, 8), tenor_two_musicmaker_three],
1431     [(50, 8), (52, 8), tenor_two_musicmaker_three],
1432     [(52, 8), (54, 8), tenor_two_musicmaker_two],
1433     [(54, 8), (56, 8), tenor_two_musicmaker_two],
1434     [(56, 8), (58, 8), tenor_two_musicmaker_two],
1435     [(58, 8), (60, 8), tenor_two_musicmaker_two],
1436     [(60, 8), (62, 8), tenor_two_musicmaker_two],
1437     [(62, 8), (64, 8), tenor_two_musicmaker_three],
1438     [(64, 8), (66, 8), tenor_two_musicmaker_three],
1439     [(66, 8), (68, 8), tenor_two_musicmaker_three],
1440     [(68, 8), (70, 8), tenor_two_musicmaker_three],
1441     [(70, 8), (72, 8), tenor_two_musicmaker_three],
1442     [(72, 8), (74, 8), tenor_two_musicmaker_two],
1443     [(74, 8), (76, 8), tenor_two_musicmaker_two],
1444     [(76, 8), (78, 8), tenor_two_musicmaker_two],
1445     [(78, 8), (80, 8), tenor_two_musicmaker_two],
1446     [(80, 8), (82, 8), tenor_two_musicmaker_two],
1447     [(82, 8), (84, 8), tenor_two_musicmaker_two],
1448     [(84, 8), (86, 8), tenor_two_musicmaker_two],
1449     [(86, 8), (88, 8), tenor_two_musicmaker_three],
1450     [(88, 8), (90, 8), tenor_two_musicmaker_three],
1451     [(90, 8), (92, 8), tenor_two_musicmaker_three],
1452     [(92, 8), (94, 8), tenor_two_musicmaker_three],
1453     [(94, 8), (96, 8), tenor_two_musicmaker_three],
1454 ]
1455 ])
1456
1457 voice_13_timespan_list = abjad.TimespanList([
1458     abjad.AnnotatedTimespan(
1459         start_offset=start_offset,
1460         stop_offset=stop_offset,
1461         annotation=MusicSpecifier(
1462             music_maker=music_maker,
1463             voice_name='Voice 13',
1464         ),
1465     )
1466     for start_offset, stop_offset, music_maker in [
1467         [(0, 8), (2, 8), tenor_three_musicmaker_three],
1468         [(2, 8), (4, 8), tenor_three_musicmaker_three],
1469         [(4, 8), (6, 8), tenor_three_musicmaker_three],
1470         [(6, 8), (8, 8), tenor_three_musicmaker_three],
1471         [(8, 8), (10, 8), tenor_three_musicmaker_three],
1472         [(10, 8), (12, 8), tenor_three_musicmaker_three],
1473         [(12, 8), (14, 8), tenor_three_musicmaker_two],

```

```

1474     [(14, 8), (16, 8), tenor_three_musicmaker_two],
1475     [(16, 8), (18, 8), tenor_three_musicmaker_two],
1476     [(18, 8), (20, 8), tenor_three_musicmaker_two],
1477     [(20, 8), (22, 8), tenor_three_musicmaker_two],
1478     [(22, 8), (24, 8), tenor_three_musicmaker_two],
1479     [(24, 8), (26, 8), tenor_three_musicmaker_three],
1480     [(26, 8), (28, 8), tenor_three_musicmaker_three],
1481     [(28, 8), (30, 8), tenor_three_musicmaker_three],
1482     [(30, 8), (32, 8), tenor_three_musicmaker_two],
1483     [(32, 8), (34, 8), tenor_three_musicmaker_two],
1484     [(34, 8), (36, 8), tenor_three_musicmaker_two],
1485     [(36, 8), (38, 8), tenor_three_musicmaker_two],
1486     [(38, 8), (40, 8), tenor_three_musicmaker_three],
1487     [(40, 8), (42, 8), tenor_three_musicmaker_three],
1488     [(42, 8), (44, 8), tenor_three_musicmaker_three],
1489     [(44, 8), (46, 8), tenor_three_musicmaker_three],
1490     [(46, 8), (48, 8), tenor_three_musicmaker_three],
1491     [(48, 8), (50, 8), tenor_three_musicmaker_three],
1492     [(50, 8), (52, 8), tenor_three_musicmaker_three],
1493     [(52, 8), (54, 8), tenor_three_musicmaker_two],
1494     [(54, 8), (56, 8), tenor_three_musicmaker_two],
1495     [(56, 8), (58, 8), tenor_three_musicmaker_two],
1496     [(58, 8), (60, 8), tenor_three_musicmaker_two],
1497     [(60, 8), (62, 8), tenor_three_musicmaker_two],
1498     [(62, 8), (64, 8), tenor_three_musicmaker_three],
1499     [(64, 8), (66, 8), tenor_three_musicmaker_three],
1500     [(66, 8), (68, 8), tenor_three_musicmaker_three],
1501     [(68, 8), (70, 8), tenor_three_musicmaker_three],
1502     [(70, 8), (72, 8), tenor_three_musicmaker_three],
1503     [(72, 8), (74, 8), tenor_three_musicmaker_two],
1504     [(74, 8), (76, 8), tenor_three_musicmaker_two],
1505     [(76, 8), (78, 8), tenor_three_musicmaker_two],
1506     [(78, 8), (80, 8), tenor_three_musicmaker_two],
1507     [(80, 8), (82, 8), tenor_three_musicmaker_two],
1508     [(82, 8), (84, 8), tenor_three_musicmaker_two],
1509     [(84, 8), (86, 8), tenor_three_musicmaker_two],
1510     [(86, 8), (88, 8), tenor_three_musicmaker_three],
1511     [(88, 8), (90, 8), tenor_three_musicmaker_three],
1512     [(90, 8), (92, 8), tenor_three_musicmaker_three],
1513     [(92, 8), (94, 8), tenor_three_musicmaker_three],
1514     [(94, 8), (96, 8), tenor_three_musicmaker_two],
1515 ]
1516 ])
1517
1518 voice_14_timespan_list = abjad.TimespanList([
1519     abjad.AnnotatedTimespan(
1520         start_offset=start_offset,
1521         stop_offset=stop_offset,
1522         annotation=MusicSpecifier(
1523             music_maker=music_maker,
1524             voice_name='Voice 14',
1525         ),
1526     )
1527     for start_offset, stop_offset, music_maker in [
1528         [(0, 8), (2, 8), tenor_four_musicmaker_two],
1529         [(2, 8), (4, 8), tenor_four_musicmaker_two],
1530         [(4, 8), (6, 8), tenor_four_musicmaker_two],
1531         [(6, 8), (8, 8), tenor_four_musicmaker_two],
1532         [(8, 8), (10, 8), tenor_four_musicmaker_two],

```

```

1533     [(10, 8), (12, 8), tenor_four_musicmaker_two],
1534     [(12, 8), (14, 8), tenor_four_musicmaker_two],
1535     [(14, 8), (16, 8), tenor_four_musicmaker_two],
1536     [(16, 8), (18, 8), tenor_four_musicmaker_three],
1537     [(18, 8), (20, 8), tenor_four_musicmaker_three],
1538     [(20, 8), (22, 8), tenor_four_musicmaker_three],
1539     [(22, 8), (24, 8), tenor_four_musicmaker_three],
1540     [(24, 8), (26, 8), tenor_four_musicmaker_three],
1541     [(26, 8), (28, 8), tenor_four_musicmaker_three],
1542     [(28, 8), (30, 8), tenor_four_musicmaker_three],
1543     [(30, 8), (32, 8), tenor_four_musicmaker_three],
1544     [(32, 8), (34, 8), tenor_four_musicmaker_two],
1545     [(34, 8), (36, 8), tenor_four_musicmaker_two],
1546     [(36, 8), (38, 8), tenor_four_musicmaker_two],
1547     [(38, 8), (40, 8), tenor_four_musicmaker_two],
1548     [(40, 8), (42, 8), tenor_four_musicmaker_two],
1549     [(42, 8), (44, 8), tenor_four_musicmaker_two],
1550     [(44, 8), (46, 8), tenor_four_musicmaker_two],
1551     [(46, 8), (48, 8), tenor_four_musicmaker_two],
1552     [(48, 8), (50, 8), tenor_four_musicmaker_three],
1553     [(50, 8), (52, 8), tenor_four_musicmaker_three],
1554     [(52, 8), (54, 8), tenor_four_musicmaker_three],
1555     [(54, 8), (56, 8), tenor_four_musicmaker_three],
1556     [(56, 8), (58, 8), tenor_four_musicmaker_three],
1557     [(58, 8), (60, 8), tenor_four_musicmaker_three],
1558     [(60, 8), (62, 8), tenor_four_musicmaker_three],
1559     [(62, 8), (64, 8), tenor_four_musicmaker_two],
1560     [(64, 8), (66, 8), tenor_four_musicmaker_two],
1561     [(66, 8), (68, 8), tenor_four_musicmaker_two],
1562     [(68, 8), (70, 8), tenor_four_musicmaker_two],
1563     [(70, 8), (72, 8), tenor_four_musicmaker_two],
1564     [(72, 8), (74, 8), tenor_four_musicmaker_two],
1565     [(74, 8), (76, 8), tenor_four_musicmaker_three],
1566     [(76, 8), (78, 8), tenor_four_musicmaker_three],
1567     [(78, 8), (80, 8), tenor_four_musicmaker_three],
1568     [(80, 8), (82, 8), tenor_four_musicmaker_three],
1569     [(82, 8), (84, 8), tenor_four_musicmaker_three],
1570     [(84, 8), (86, 8), tenor_four_musicmaker_two],
1571     [(86, 8), (88, 8), tenor_four_musicmaker_two],
1572     [(88, 8), (90, 8), tenor_four_musicmaker_two],
1573     [(90, 8), (92, 8), tenor_four_musicmaker_two],
1574     [(92, 8), (94, 8), tenor_four_musicmaker_three],
1575     [(94, 8), (96, 8), tenor_four_musicmaker_three],
1576   ],
1577 ])
1578
1579 voice_15_timespan_list = abjad.TimespanList([
1580     abjad.AnnotatedTimespan(
1581         start_offset=start_offset,
1582         stop_offset=stop_offset,
1583         annotation=MusicSpecifier(
1584             music_maker=music_maker,
1585             voice_name='Voice 15',
1586         ),
1587     )
1588     for start_offset, stop_offset, music_maker in [
1589         [(0, 8), (2, 8), tenor_five_musicmaker_two],
1590         [(2, 8), (4, 8), tenor_five_musicmaker_two],
1591         [(4, 8), (6, 8), tenor_five_musicmaker_two],

```

```

1592     [(6, 8), (8, 8), tenor_five_musicmaker_three],
1593     [(8, 8), (10, 8), tenor_five_musicmaker_three],
1594     [(10, 8), (12, 8), tenor_five_musicmaker_three],
1595     [(12, 8), (14, 8), tenor_five_musicmaker_two],
1596     [(14, 8), (16, 8), tenor_five_musicmaker_two],
1597     [(16, 8), (18, 8), tenor_five_musicmaker_two],
1598     [(18, 8), (20, 8), tenor_five_musicmaker_two],
1599     [(20, 8), (22, 8), tenor_five_musicmaker_three],
1600     [(22, 8), (24, 8), tenor_five_musicmaker_three],
1601     [(24, 8), (26, 8), tenor_five_musicmaker_three],
1602     [(26, 8), (28, 8), tenor_five_musicmaker_three],
1603     [(28, 8), (30, 8), tenor_five_musicmaker_two],
1604     [(30, 8), (32, 8), tenor_five_musicmaker_two],
1605     [(32, 8), (34, 8), tenor_five_musicmaker_two],
1606     [(34, 8), (36, 8), tenor_five_musicmaker_three],
1607     [(36, 8), (38, 8), tenor_five_musicmaker_three],
1608     [(38, 8), (40, 8), tenor_five_musicmaker_three],
1609     [(40, 8), (42, 8), tenor_five_musicmaker_three],
1610     [(42, 8), (44, 8), tenor_five_musicmaker_two],
1611     [(44, 8), (46, 8), tenor_five_musicmaker_two],
1612     [(46, 8), (48, 8), tenor_five_musicmaker_three],
1613     [(48, 8), (50, 8), tenor_five_musicmaker_three],
1614     [(50, 8), (52, 8), tenor_five_musicmaker_three],
1615     [(52, 8), (54, 8), tenor_five_musicmaker_three],
1616     [(54, 8), (56, 8), tenor_five_musicmaker_three],
1617     [(56, 8), (58, 8), tenor_five_musicmaker_three],
1618     [(58, 8), (60, 8), tenor_five_musicmaker_two],
1619     [(60, 8), (62, 8), tenor_five_musicmaker_three],
1620     [(62, 8), (64, 8), tenor_five_musicmaker_three],
1621     [(64, 8), (66, 8), tenor_five_musicmaker_three],
1622     [(66, 8), (68, 8), tenor_five_musicmaker_three],
1623     [(68, 8), (70, 8), tenor_five_musicmaker_three],
1624     [(70, 8), (72, 8), tenor_five_musicmaker_three],
1625     [(72, 8), (74, 8), tenor_five_musicmaker_three],
1626     [(74, 8), (76, 8), tenor_five_musicmaker_two],
1627     [(76, 8), (78, 8), tenor_five_musicmaker_two],
1628     [(78, 8), (80, 8), tenor_five_musicmaker_two],
1629     [(80, 8), (82, 8), tenor_five_musicmaker_two],
1630     [(82, 8), (84, 8), tenor_five_musicmaker_two],
1631     [(84, 8), (86, 8), tenor_five_musicmaker_two],
1632     [(86, 8), (88, 8), tenor_five_musicmaker_two],
1633     [(88, 8), (90, 8), tenor_five_musicmaker_two],
1634     [(90, 8), (92, 8), tenor_five_musicmaker_three],
1635     [(92, 8), (94, 8), tenor_five_musicmaker_three],
1636     [(94, 8), (96, 8), tenor_five_musicmaker_three],
1637 ]
1638 ])
1639
1640 voice_16_timespan_list = abjad.TimespanList([
1641     abjad.AnnotatedTimespan(
1642         start_offset=start_offset,
1643         stop_offset=stop_offset,
1644         annotation=MusicSpecifier(
1645             music_maker=music_maker,
1646             voice_name='Voice 16',
1647         ),
1648     )
1649     for start_offset, stop_offset, music_maker in [
1650         [(0, 8), (2, 8), baritone_one_musicmaker_two],

```

```

1651     [(2, 8), (4, 8), baritone_one_musicmaker_two],
1652     [(4, 8), (6, 8), baritone_one_musicmaker_two],
1653     [(6, 8), (8, 8), baritone_one_musicmaker_two],
1654     [(8, 8), (10, 8), baritone_one_musicmaker_three],
1655     [(10, 8), (12, 8), baritone_one_musicmaker_three],
1656     [(12, 8), (14, 8), baritone_one_musicmaker_three],
1657     [(14, 8), (16, 8), baritone_one_musicmaker_two],
1658     [(16, 8), (18, 8), baritone_one_musicmaker_two],
1659     [(18, 8), (20, 8), baritone_one_musicmaker_two],
1660     [(20, 8), (22, 8), baritone_one_musicmaker_two],
1661     [(22, 8), (24, 8), baritone_one_musicmaker_three],
1662     [(24, 8), (26, 8), baritone_one_musicmaker_three],
1663     [(26, 8), (28, 8), baritone_one_musicmaker_three],
1664     [(28, 8), (30, 8), baritone_one_musicmaker_two],
1665     [(30, 8), (32, 8), baritone_one_musicmaker_two],
1666     [(32, 8), (34, 8), baritone_one_musicmaker_two],
1667     [(34, 8), (36, 8), baritone_one_musicmaker_two],
1668     [(36, 8), (38, 8), baritone_one_musicmaker_three],
1669     [(38, 8), (40, 8), baritone_one_musicmaker_three],
1670     [(40, 8), (42, 8), baritone_one_musicmaker_three],
1671     [(42, 8), (44, 8), baritone_one_musicmaker_two],
1672     [(44, 8), (46, 8), baritone_one_musicmaker_two],
1673     [(46, 8), (48, 8), baritone_one_musicmaker_two],
1674     [(48, 8), (50, 8), baritone_one_musicmaker_two],
1675     [(50, 8), (52, 8), baritone_one_musicmaker_three],
1676     [(52, 8), (54, 8), baritone_one_musicmaker_three],
1677     [(54, 8), (56, 8), baritone_one_musicmaker_three],
1678     [(56, 8), (58, 8), baritone_one_musicmaker_two],
1679     [(58, 8), (60, 8), baritone_one_musicmaker_two],
1680     [(60, 8), (62, 8), baritone_one_musicmaker_two],
1681     [(62, 8), (64, 8), baritone_one_musicmaker_two],
1682     [(64, 8), (66, 8), baritone_one_musicmaker_three],
1683     [(66, 8), (68, 8), baritone_one_musicmaker_three],
1684     [(68, 8), (70, 8), baritone_one_musicmaker_three],
1685     [(70, 8), (72, 8), baritone_one_musicmaker_two],
1686     [(72, 8), (74, 8), baritone_one_musicmaker_two],
1687     [(74, 8), (76, 8), baritone_one_musicmaker_two],
1688     [(76, 8), (78, 8), baritone_one_musicmaker_two],
1689     [(78, 8), (80, 8), baritone_one_musicmaker_three],
1690     [(80, 8), (82, 8), baritone_one_musicmaker_three],
1691     [(82, 8), (84, 8), baritone_one_musicmaker_three],
1692     [(84, 8), (86, 8), baritone_one_musicmaker_two],
1693     [(86, 8), (88, 8), baritone_one_musicmaker_two],
1694     [(88, 8), (90, 8), baritone_one_musicmaker_two],
1695     [(90, 8), (92, 8), baritone_one_musicmaker_two],
1696     [(92, 8), (94, 8), baritone_one_musicmaker_three],
1697     [(94, 8), (96, 8), baritone_one_musicmaker_three],
1698   ]
1699 ])
1700
1701 voice_17_timespan_list = abjad.TimespanList([
1702     abjad.AnnotatedTimespan(
1703         start_offset=start_offset,
1704         stop_offset=stop_offset,
1705         annotation=MusicSpecifier(
1706             music_maker=music_maker,
1707             voice_name='Voice 17',
1708         ),
1709     )

```

```

1710     for start_offset, stop_offset, music_maker in [
1711         [(0, 8), (2, 8), baritone_two_musicmaker_three],
1712         [(2, 8), (4, 8), baritone_two_musicmaker_three],
1713         [(4, 8), (6, 8), baritone_two_musicmaker_two],
1714         [(6, 8), (8, 8), baritone_two_musicmaker_two],
1715         [(8, 8), (10, 8), baritone_two_musicmaker_two],
1716         [(10, 8), (12, 8), baritone_two_musicmaker_three],
1717         [(12, 8), (14, 8), baritone_two_musicmaker_three],
1718         [(14, 8), (16, 8), baritone_two_musicmaker_three],
1719         [(16, 8), (18, 8), baritone_two_musicmaker_three],
1720         [(18, 8), (20, 8), baritone_two_musicmaker_two],
1721         [(20, 8), (22, 8), baritone_two_musicmaker_two],
1722         [(22, 8), (24, 8), baritone_two_musicmaker_two],
1723         [(24, 8), (26, 8), baritone_two_musicmaker_two],
1724         [(26, 8), (28, 8), baritone_two_musicmaker_two],
1725         [(28, 8), (30, 8), baritone_two_musicmaker_three],
1726         [(30, 8), (32, 8), baritone_two_musicmaker_three],
1727         [(32, 8), (34, 8), baritone_two_musicmaker_three],
1728         [(34, 8), (36, 8), baritone_two_musicmaker_three],
1729         [(36, 8), (38, 8), baritone_two_musicmaker_three],
1730         [(38, 8), (40, 8), baritone_two_musicmaker_three],
1731         [(40, 8), (42, 8), baritone_two_musicmaker_two],
1732         [(42, 8), (44, 8), baritone_two_musicmaker_two],
1733         [(44, 8), (46, 8), baritone_two_musicmaker_two],
1734         [(46, 8), (48, 8), baritone_two_musicmaker_two],
1735         [(48, 8), (50, 8), baritone_two_musicmaker_two],
1736         [(50, 8), (52, 8), baritone_two_musicmaker_two],
1737         [(52, 8), (54, 8), baritone_two_musicmaker_two],
1738         [(54, 8), (56, 8), baritone_two_musicmaker_three],
1739         [(56, 8), (58, 8), baritone_two_musicmaker_three],
1740         [(58, 8), (60, 8), baritone_two_musicmaker_three],
1741         [(60, 8), (62, 8), baritone_two_musicmaker_three],
1742         [(62, 8), (64, 8), baritone_two_musicmaker_three],
1743         [(64, 8), (66, 8), baritone_two_musicmaker_three],
1744         [(66, 8), (68, 8), baritone_two_musicmaker_three],
1745         [(68, 8), (70, 8), baritone_two_musicmaker_two],
1746         [(70, 8), (72, 8), baritone_two_musicmaker_two],
1747         [(72, 8), (74, 8), baritone_two_musicmaker_two],
1748         [(74, 8), (76, 8), baritone_two_musicmaker_two],
1749         [(76, 8), (78, 8), baritone_two_musicmaker_two],
1750         [(78, 8), (80, 8), baritone_two_musicmaker_two],
1751         [(80, 8), (82, 8), baritone_two_musicmaker_two],
1752         [(82, 8), (84, 8), baritone_two_musicmaker_three],
1753         [(84, 8), (86, 8), baritone_two_musicmaker_three],
1754         [(86, 8), (88, 8), baritone_two_musicmaker_three],
1755         [(88, 8), (90, 8), baritone_two_musicmaker_three],
1756         [(90, 8), (92, 8), baritone_two_musicmaker_three],
1757         [(92, 8), (94, 8), baritone_two_musicmaker_three],
1758         [(94, 8), (96, 8), baritone_two_musicmaker_three],
1759     ]
1760 ])
1761
1762 voice_18_timespan_list = abjad.TimespanList([
1763     abjad.AnnotatedTimespan(
1764         start_offset=start_offset,
1765         stop_offset=stop_offset,
1766         annotation=MusicSpecifier(
1767             music_maker=music_maker,
1768             voice_name='Voice 18',

```

```

1769     ),
1770 )
1771 for start_offset, stop_offset, music_maker in [
1772     [(0, 8), (2, 8), baritone_three_musicmaker_three],
1773     [(2, 8), (4, 8), baritone_three_musicmaker_three],
1774     [(4, 8), (6, 8), baritone_three_musicmaker_three],
1775     [(6, 8), (8, 8), baritone_three_musicmaker_three],
1776     [(8, 8), (10, 8), baritone_three_musicmaker_two],
1777     [(10, 8), (12, 8), baritone_three_musicmaker_two],
1778     [(12, 8), (14, 8), baritone_three_musicmaker_two],
1779     [(14, 8), (16, 8), baritone_three_musicmaker_two],
1780     [(16, 8), (18, 8), baritone_three_musicmaker_three],
1781     [(18, 8), (20, 8), baritone_three_musicmaker_three],
1782     [(20, 8), (22, 8), baritone_three_musicmaker_three],
1783     [(22, 8), (24, 8), baritone_three_musicmaker_three],
1784     [(24, 8), (26, 8), baritone_three_musicmaker_two],
1785     [(26, 8), (28, 8), baritone_three_musicmaker_two],
1786     [(28, 8), (30, 8), baritone_three_musicmaker_two],
1787     [(30, 8), (32, 8), baritone_three_musicmaker_two],
1788     [(32, 8), (34, 8), baritone_three_musicmaker_three],
1789     [(34, 8), (36, 8), baritone_three_musicmaker_three],
1790     [(36, 8), (38, 8), baritone_three_musicmaker_three],
1791     [(38, 8), (40, 8), baritone_three_musicmaker_three],
1792     [(40, 8), (42, 8), baritone_three_musicmaker_two],
1793     [(42, 8), (44, 8), baritone_three_musicmaker_two],
1794     [(44, 8), (46, 8), baritone_three_musicmaker_two],
1795     [(46, 8), (48, 8), baritone_three_musicmaker_two],
1796     [(48, 8), (50, 8), baritone_three_musicmaker_three],
1797     [(50, 8), (52, 8), baritone_three_musicmaker_three],
1798     [(52, 8), (54, 8), baritone_three_musicmaker_three],
1799     [(54, 8), (56, 8), baritone_three_musicmaker_three],
1800     [(56, 8), (58, 8), baritone_three_musicmaker_two],
1801     [(58, 8), (60, 8), baritone_three_musicmaker_two],
1802     [(60, 8), (62, 8), baritone_three_musicmaker_two],
1803     [(62, 8), (64, 8), baritone_three_musicmaker_two],
1804     [(64, 8), (66, 8), baritone_three_musicmaker_three],
1805     [(66, 8), (68, 8), baritone_three_musicmaker_three],
1806     [(68, 8), (70, 8), baritone_three_musicmaker_three],
1807     [(70, 8), (72, 8), baritone_three_musicmaker_three],
1808     [(72, 8), (74, 8), baritone_three_musicmaker_two],
1809     [(74, 8), (76, 8), baritone_three_musicmaker_two],
1810     [(76, 8), (78, 8), baritone_three_musicmaker_two],
1811     [(78, 8), (80, 8), baritone_three_musicmaker_two],
1812     [(80, 8), (82, 8), baritone_three_musicmaker_three],
1813     [(82, 8), (84, 8), baritone_three_musicmaker_three],
1814     [(84, 8), (86, 8), baritone_three_musicmaker_three],
1815     [(86, 8), (88, 8), baritone_three_musicmaker_three],
1816     [(88, 8), (90, 8), baritone_three_musicmaker_two],
1817     [(90, 8), (92, 8), baritone_three_musicmaker_two],
1818     [(92, 8), (94, 8), baritone_three_musicmaker_two],
1819     [(94, 8), (96, 8), baritone_three_musicmaker_two],
1820 ]
1821 ])
1822
1823 voice_19_timespan_list = abjad.TimespanList([
1824     abjad.AnnotatedTimespan(
1825         start_offset=start_offset,
1826         stop_offset=stop_offset,
1827         annotation=MusicSpecifier(

```

```

1828     music_maker=music_maker,
1829     voice_name='Voice 19',
1830   ),
1831 )
1832 for start_offset, stop_offset, music_maker in [
1833   [(0, 8), (2, 8), bass_one_musicmaker_two],
1834   [(2, 8), (4, 8), bass_one_musicmaker_two],
1835   [(4, 8), (6, 8), bass_one_musicmaker_two],
1836   [(6, 8), (8, 8), bass_one_musicmaker_three],
1837   [(8, 8), (10, 8), bass_one_musicmaker_three],
1838   [(10, 8), (12, 8), bass_one_musicmaker_two],
1839   [(12, 8), (14, 8), bass_one_musicmaker_two],
1840   [(14, 8), (16, 8), bass_one_musicmaker_two],
1841   [(16, 8), (18, 8), bass_one_musicmaker_two],
1842   [(18, 8), (20, 8), bass_one_musicmaker_three],
1843   [(20, 8), (22, 8), bass_one_musicmaker_two],
1844   [(22, 8), (24, 8), bass_one_musicmaker_two],
1845   [(24, 8), (26, 8), bass_one_musicmaker_two],
1846   [(26, 8), (28, 8), bass_one_musicmaker_two],
1847   [(28, 8), (30, 8), bass_one_musicmaker_two],
1848   [(30, 8), (32, 8), bass_one_musicmaker_three],
1849   [(32, 8), (34, 8), bass_one_musicmaker_three],
1850   [(34, 8), (36, 8), bass_one_musicmaker_three],
1851   [(36, 8), (38, 8), bass_one_musicmaker_three],
1852   [(38, 8), (40, 8), bass_one_musicmaker_three],
1853   [(40, 8), (42, 8), bass_one_musicmaker_three],
1854   [(42, 8), (44, 8), bass_one_musicmaker_three],
1855   [(44, 8), (46, 8), bass_one_musicmaker_three],
1856   [(46, 8), (48, 8), bass_one_musicmaker_two],
1857   [(48, 8), (50, 8), bass_one_musicmaker_two],
1858   [(50, 8), (52, 8), bass_one_musicmaker_two],
1859   [(52, 8), (54, 8), bass_one_musicmaker_two],
1860   [(54, 8), (56, 8), bass_one_musicmaker_two],
1861   [(56, 8), (58, 8), bass_one_musicmaker_two],
1862   [(58, 8), (60, 8), bass_one_musicmaker_three],
1863   [(60, 8), (62, 8), bass_one_musicmaker_three],
1864   [(62, 8), (64, 8), bass_one_musicmaker_three],
1865   [(64, 8), (68, 8), bass_one_musicmaker_three],
1866   [(68, 8), (70, 8), bass_one_musicmaker_three],
1867   [(70, 8), (72, 8), bass_one_musicmaker_three],
1868   [(72, 8), (74, 8), bass_one_musicmaker_three],
1869   [(74, 8), (76, 8), bass_one_musicmaker_two],
1870   [(76, 8), (78, 8), bass_one_musicmaker_two],
1871   [(78, 8), (80, 8), bass_one_musicmaker_two],
1872   [(80, 8), (82, 8), bass_one_musicmaker_two],
1873   [(82, 8), (84, 8), bass_one_musicmaker_two],
1874   [(84, 8), (86, 8), bass_one_musicmaker_two],
1875   [(86, 8), (88, 8), bass_one_musicmaker_two],
1876   [(88, 8), (90, 8), bass_one_musicmaker_three],
1877   [(90, 8), (92, 8), bass_one_musicmaker_three],
1878   [(92, 8), (94, 8), bass_one_musicmaker_three],
1879   [(94, 8), (96, 8), bass_one_musicmaker_three],
1880 ]
1881 ])
1882
1883 voice_20_timespan_list = abjad.TimespanList([
1884     abjad.AnnotatedTimespan(
1885         start_offset=start_offset,
1886         stop_offset=stop_offset,

```

```

1887     annotation=MusicSpecifier(
1888         music_maker=music_maker,
1889         voice_name='Voice 20',
1890     ),
1891 )
1892 for start_offset, stop_offset, music_maker in [
1893     [(0, 8), (2, 8), bass_two_musicmaker_two],
1894     [(2, 8), (4, 8), bass_two_musicmaker_two],
1895     [(4, 8), (6, 8), bass_two_musicmaker_two],
1896     [(6, 8), (8, 8), bass_two_musicmaker_three],
1897     [(8, 8), (10, 8), bass_two_musicmaker_three],
1898     [(10, 8), (12, 8), bass_two_musicmaker_two],
1899     [(12, 8), (14, 8), bass_two_musicmaker_two],
1900     [(14, 8), (16, 8), bass_two_musicmaker_two],
1901     [(16, 8), (18, 8), bass_two_musicmaker_two],
1902     [(18, 8), (20, 8), bass_two_musicmaker_three],
1903     [(20, 8), (22, 8), bass_two_musicmaker_two],
1904     [(22, 8), (24, 8), bass_two_musicmaker_two],
1905     [(24, 8), (26, 8), bass_two_musicmaker_two],
1906     [(26, 8), (28, 8), bass_two_musicmaker_two],
1907     [(28, 8), (30, 8), bass_two_musicmaker_two],
1908     [(30, 8), (32, 8), bass_two_musicmaker_three],
1909     [(32, 8), (34, 8), bass_two_musicmaker_three],
1910     [(34, 8), (36, 8), bass_two_musicmaker_three],
1911     [(36, 8), (38, 8), bass_two_musicmaker_three],
1912     [(38, 8), (40, 8), bass_two_musicmaker_three],
1913     [(40, 8), (42, 8), bass_two_musicmaker_three],
1914     [(42, 8), (44, 8), bass_two_musicmaker_three],
1915     [(44, 8), (46, 8), bass_two_musicmaker_three],
1916     [(46, 8), (48, 8), bass_two_musicmaker_two],
1917     [(48, 8), (50, 8), bass_two_musicmaker_two],
1918     [(50, 8), (52, 8), bass_two_musicmaker_two],
1919     [(52, 8), (54, 8), bass_two_musicmaker_two],
1920     [(54, 8), (56, 8), bass_two_musicmaker_two],
1921     [(56, 8), (58, 8), bass_two_musicmaker_two],
1922     [(58, 8), (60, 8), bass_two_musicmaker_three],
1923     [(60, 8), (62, 8), bass_two_musicmaker_three],
1924     [(62, 8), (64, 8), bass_two_musicmaker_three],
1925     [(64, 8), (68, 8), bass_two_musicmaker_three],
1926     [(68, 8), (70, 8), bass_two_musicmaker_three],
1927     [(70, 8), (72, 8), bass_two_musicmaker_three],
1928     [(72, 8), (74, 8), bass_two_musicmaker_three],
1929     [(74, 8), (76, 8), bass_two_musicmaker_two],
1930     [(76, 8), (78, 8), bass_two_musicmaker_two],
1931     [(78, 8), (80, 8), bass_two_musicmaker_two],
1932     [(80, 8), (82, 8), bass_two_musicmaker_two],
1933     [(82, 8), (84, 8), bass_two_musicmaker_two],
1934     [(84, 8), (86, 8), bass_two_musicmaker_two],
1935     [(86, 8), (88, 8), bass_two_musicmaker_two],
1936     [(88, 8), (90, 8), bass_two_musicmaker_three],
1937     [(90, 8), (92, 8), bass_two_musicmaker_three],
1938     [(92, 8), (94, 8), bass_two_musicmaker_three],
1939     [(94, 8), (96, 8), bass_two_musicmaker_three],
1940 ]
1941 ])
1942
1943 voice_21_timespan_list = abjad.TimespanList([
1944     abjad.AnnotatedTimespan(
1945         start_offset=start_offset,

```

```

1946     stop_offset=stop_offset,
1947     annotation=MusicSpecifier(
1948         music_maker=music_maker,
1949         voice_name='Voice 21',
1950     ),
1951 )
1952 for start_offset, stop_offset, music_maker in [
1953     [(0, 8), (2, 8), contrabass_musicmaker_two],
1954     [(2, 8), (4, 8), contrabass_musicmaker_two],
1955     [(4, 8), (6, 8), contrabass_musicmaker_two],
1956     [(6, 8), (8, 8), contrabass_musicmaker_three],
1957     [(8, 8), (10, 8), contrabass_musicmaker_three],
1958     [(10, 8), (12, 8), contrabass_musicmaker_two],
1959     [(12, 8), (14, 8), contrabass_musicmaker_two],
1960     [(14, 8), (16, 8), contrabass_musicmaker_two],
1961     [(16, 8), (18, 8), contrabass_musicmaker_two],
1962     [(18, 8), (20, 8), contrabass_musicmaker_three],
1963     [(20, 8), (22, 8), contrabass_musicmaker_two],
1964     [(22, 8), (24, 8), contrabass_musicmaker_two],
1965     [(24, 8), (26, 8), contrabass_musicmaker_two],
1966     [(26, 8), (28, 8), contrabass_musicmaker_two],
1967     [(28, 8), (30, 8), contrabass_musicmaker_two],
1968     [(30, 8), (32, 8), contrabass_musicmaker_three],
1969     [(32, 8), (34, 8), contrabass_musicmaker_three],
1970     [(34, 8), (36, 8), contrabass_musicmaker_three],
1971     [(36, 8), (38, 8), contrabass_musicmaker_three],
1972     [(38, 8), (40, 8), contrabass_musicmaker_three],
1973     [(40, 8), (42, 8), contrabass_musicmaker_three],
1974     [(42, 8), (44, 8), contrabass_musicmaker_three],
1975     [(44, 8), (46, 8), contrabass_musicmaker_three],
1976     [(46, 8), (48, 8), contrabass_musicmaker_two],
1977     [(48, 8), (50, 8), contrabass_musicmaker_two],
1978     [(50, 8), (52, 8), contrabass_musicmaker_two],
1979     [(52, 8), (54, 8), contrabass_musicmaker_two],
1980     [(54, 8), (56, 8), contrabass_musicmaker_two],
1981     [(56, 8), (58, 8), contrabass_musicmaker_two],
1982     [(58, 8), (60, 8), contrabass_musicmaker_three],
1983     [(60, 8), (62, 8), contrabass_musicmaker_three],
1984     [(62, 8), (64, 8), contrabass_musicmaker_three],
1985     [(64, 8), (68, 8), contrabass_musicmaker_three],
1986     [(68, 8), (70, 8), contrabass_musicmaker_three],
1987     [(70, 8), (72, 8), contrabass_musicmaker_three],
1988     [(72, 8), (74, 8), contrabass_musicmaker_three],
1989     [(74, 8), (76, 8), contrabass_musicmaker_two],
1990     [(76, 8), (78, 8), contrabass_musicmaker_two],
1991     [(78, 8), (80, 8), contrabass_musicmaker_two],
1992     [(80, 8), (82, 8), contrabass_musicmaker_two],
1993     [(82, 8), (84, 8), contrabass_musicmaker_two],
1994     [(84, 8), (86, 8), contrabass_musicmaker_two],
1995     [(86, 8), (88, 8), contrabass_musicmaker_two],
1996     [(88, 8), (90, 8), contrabass_musicmaker_three],
1997     [(90, 8), (92, 8), contrabass_musicmaker_three],
1998     [(92, 8), (94, 8), contrabass_musicmaker_three],
1999     [(94, 8), (96, 8), contrabass_musicmaker_three],
2000 ]
2001 ])
2002
2003 all_timespan_lists = {
2004     'Voice 1': voice_1_timespan_list,

```

```

2005     'Voice 2': voice_2_timespan_list,
2006     'Voice 3': voice_3_timespan_list,
2007     'Voice 4': voice_4_timespan_list,
2008     'Voice 5': voice_5_timespan_list,
2009     'Voice 6': voice_6_timespan_list,
2010     'Voice 7': voice_7_timespan_list,
2011     'Voice 8': voice_8_timespan_list,
2012     'Voice 9': voice_9_timespan_list,
2013     'Voice 10': voice_10_timespan_list,
2014     'Voice 11': voice_11_timespan_list,
2015     'Voice 12': voice_12_timespan_list,
2016     'Voice 13': voice_13_timespan_list,
2017     'Voice 14': voice_14_timespan_list,
2018     'Voice 15': voice_15_timespan_list,
2019     'Voice 16': voice_16_timespan_list,
2020     'Voice 17': voice_17_timespan_list,
2021     'Voice 18': voice_18_timespan_list,
2022     'Voice 19': voice_19_timespan_list,
2023     'Voice 20': voice_20_timespan_list,
2024     'Voice 21': voice_21_timespan_list,
2025 }
2026
2027 global_timespan = abjad.Timespan(
2028     start_offset=0,
2029     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
2030 )
2031
2032 for voice_name, timespan_list in all_timespan_lists.items():
2033     silences = abjad.TimespanList([global_timespan])
2034     silences.extend(timespan_list)
2035     silences.sort()
2036     silences.compute_logical_xor()
2037     for silence_timespan in silences:
2038         timespan_list.append(
2039             abjad.AnnotatedTimespan(
2040                 start_offset=silence_timespan.start_offset,
2041                 stop_offset=silence_timespan.stop_offset,
2042                 annotation=MusicSpecifier(
2043                     music_maker=None,
2044                     voice_name=voice_name,
2045                 ),
2046             )
2047         )
2048     timespan_list.sort()
2049
2050 for voice_name, timespan_list in all_timespan_lists.items():
2051     shards = timespan_list.split_at_offsets(bounds)
2052     split_timespan_list = abjad.TimespanList()
2053     for shard in shards:
2054         split_timespan_list.extend(shard)
2055     split_timespan_list.sort()
2056     all_timespan_lists[voice_name] = timespan_list
2057
2058 score = abjad.Score([
2059     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
2060     abjad.StaffGroup(
2061         [
2062             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
2063             lilypond_type='Staff',),

```

```

2063     abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
2064         lilypond_type='Staff',),
2065             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
2066                 lilypond_type='Staff',),
2067                     abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
2068                         lilypond_type='Staff',),
2069                             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
2070                                 lilypond_type='Staff',),
2071                                     abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
2072                                         lilypond_type='Staff',),
2073                                             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
2074                                                 lilypond_type='Staff',),
2075                                                     abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
2076                                                         lilypond_type='Staff',),
2077                                                             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
2078                                                                 lilypond_type='Staff',),
2079                     abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
2080                         lilypond_type='Staff',),
2081                             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
2082                                 lilypond_type='Staff',),
2083                     abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
2084                         lilypond_type='Staff',),
2085                             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
2086                                 lilypond_type='Staff',),
2087                     abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
2088                         lilypond_type='Staff',),
2089                             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
2090                                 lilypond_type='Staff',),
2091                     abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
2092                         lilypond_type='Staff',),
2093                             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
2094                                 lilypond_type='Staff',),
2095                     abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
2096                         lilypond_type='Staff',),
2097                             abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
2098                                 lilypond_type='Staff',),
2099                     abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
2100                         lilypond_type='Staff',),
2101                             abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
2102                                 lilypond_type='Staff',),
2103                         ],
2104                             name='Staff Group',
2105 )
2106 ])
2107
2108 for time_signature in time_signatures:
2109     skip = abjad.Skip(1, multiplier=(time_signature))
2110     abjad.attach(time_signature, skip)
2111     score['Global Context'].append(skip)
2112
2113 print('Making containers ...')
2114
2115 def make_container(music_maker, durations):
2116     selections = music_maker(durations)
2117     container = abjad.Container([])
2118     container.extend(selections)
2119     return container
2120
2121 def key_function(timespan):

```

```

2102     return timespan.annotation.music_maker or silence_maker
2103
2104 for voice_name, timespan_list in all_timespan_lists.items():
2105     for music_maker, grouper in itertools.groupby(
2106         timespan_list,
2107         key=key_function,
2108     ):
2109         durations = [timespan.duration for timespan in grouper]
2110         container = make_container(music_maker, durations)
2111         voice = score[voice_name]
2112         voice.append(container)
2113
2114 print('Splitting and rewriting ...')
2115 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
2116     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
2117         time_signature = time_signatures[i]
2118         abjad.mutate(shard).rewrite_meter(time_signature)
2119
2120 print('Beaming runs ...')
2121 for voice in abjad.select(score).components(abjad.Voice):
2122     for run in abjad.select(voice).runs():
2123         specifier = abjadext.rmakers.BeamSpecifier(
2124             beam_each_division=False,
2125         )
2126         specifier(run)
2127         abjad.beam(voice[:, beam_lone_notes=False, beam_rests=False])
2128
2129 print('Stopping Hairpins ...')
2130 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2131     for rest in abjad.iterate(staff).components(abjad.Rest):
2132         previous_leaf = abjad.inspect(rest).leaf(-1)
2133         if isinstance(previous_leaf, abjad.Note):
2134             abjad.attach(abjad.StopHairpin(), rest)
2135         elif isinstance(previous_leaf, abjad.Chord):
2136             abjad.attach(abjad.StopHairpin(), rest)
2137         elif isinstance(previous_leaf, abjad.Rest):
2138             pass
2139
2140 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2141     first_leaf = abjad.select(staff).leaves()[0]
2142     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
2143     stop_trill = abjad.LilyPondLiteral(r'\stopTrillSpan', format_slot='after')
2144     abjad.attach(stop, first_leaf)
2145     abjad.attach(stop_trill, first_leaf)
2146
2147 staffs = [staff for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff)]
2148
2149 scales = [
2150     soprano_random_walk_notes,
2151     soprano_1_random_walk_notes,
2152     soprano_2_random_walk_notes,
2153     soprano_3_random_walk_notes,
2154     alto_1_random_walk_notes,
2155     alto_2_random_walk_notes,
2156     alto_3_random_walk_notes,
2157     alto_4_random_walk_notes,
2158     alto_5_random_walk_notes,
2159     alto_6_random_walk_notes,
```

```

2160     tenor_1_random_walk_notes ,
2161     tenor_2_random_walk_notes ,
2162     tenor_3_random_walk_notes ,
2163     tenor_4_random_walk_notes ,
2164     tenor_5_random_walk_notes ,
2165     baritone_1_random_walk_notes ,
2166     baritone_2_random_walk_notes ,
2167     baritone_3_random_walk_notes ,
2168     bass_1_random_walk_notes ,
2169     bass_2_random_walk_notes ,
2170     contrabass_random_walk_notes ,
2171 ]
2172
2173 for staff , scale in zip(staffs , scales):
2174     logicl_ties = [i for i in abjad.iterate(staff).logical_ties(pitched=True)]
2175     pitches = cyc(scale)
2176     for i , logicl_tie in enumerate(logicl_ties):
2177         if logicl_tie.is_pitched ==True:
2178             pitch = next(pitches)
2179             for note in logicl_tie:
2180                 note.written_pitch = pitch
2181
2182 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2183     abjad.glissando(
2184         staff[:],
2185         allow_repeats=True ,
2186     )
2187
2188 print('Adding attachments ...')
2189 bar_line = abjad.BarLine('||')
2190 # metro = abjad.MetronomeMark((1, 4), 90)
2191 markup = abjad.Markup(r'\bold { G }')
2192 mark = abjad.RehearsalMark(markup=markup)
2193
2194 instruments = cyc([
2195     abjad.SopraninoSaxophone(),
2196     abjad.SopranoSaxophone(),
2197     abjad.SopranoSaxophone(),
2198     abjad.SopranoSaxophone(),
2199     abjad.AltoSaxophone(),
2200     abjad.AltoSaxophone(),
2201     abjad.AltoSaxophone(),
2202     abjad.AltoSaxophone(),
2203     abjad.AltoSaxophone(),
2204     abjad.AltoSaxophone(),
2205     abjad.TenorSaxophone(),
2206     abjad.TenorSaxophone(),
2207     abjad.TenorSaxophone(),
2208     abjad.TenorSaxophone(),
2209     abjad.TenorSaxophone(),
2210     abjad.BaritoneSaxophone(),
2211     abjad.BaritoneSaxophone(),
2212     abjad.BaritoneSaxophone(),
2213     abjad.BassSaxophone(),
2214     abjad.BassSaxophone(),
2215     abjad.ContrabassSaxophone(),
2216 ])
2217
2218 abbreviations = cyc([

```

```

2219 abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
2220 abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
2221 abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
2222 abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
2223 abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
2224 abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
2225 abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
2226 abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
2227 abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
2228 abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
2229 abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
2230 abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
2231 abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
2232 abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
2233 abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
2234 abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
2235 abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
2236 abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
2237 abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
2238 abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
2239 abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
2240 ])
2241
2242 names = cyc([
2243 abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
2244 abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
2245 abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
2246 abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
2247 abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
2248 abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
2249 abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
2250 abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
2251 abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
2252 abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
2253 abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
2254 abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
2255 abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
2256 abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
2257 abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
2258 abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
2259 abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
2260 abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
2261 abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
2262 abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
2263 abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
2264 ])
2265
2266 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2267     leaf1 = abjad.select(staff).leaves()[0]
2268     abjad.attach(next(instruments), leaf1)
2269     abjad.attach(next(abbreviations), leaf1)
2270     abjad.attach(next(names), leaf1)
2271
2272 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
2273     leaf1 = abjad.select(staff).leaves()[0]
2274     last_leaf = abjad.select(staff).leaves()[-1]
2275     abjad.attach(bar_line, last_leaf)
2276
2277 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):

```

```

2278 leaf1 = abjad.select(staff).leaves()[0]
2279 abjad.attach(mark, leaf1)
2280
2281 score_file = abjad.LilyPondFile.new(
2282     score,
2283     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2284 _stylesheets/abjad.ily'],
2285 )
2286
2287 abjad.SegmentMaker.comment_measure_numbers(score)
2288 ######
2289 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_G'
2290 pdf_path = f'{directory}/Section_G.pdf'
2291 path = pathlib.Path('Section_G.pdf')
2292 if path.exists():
2293     print(f'Removing {pdf_path} ...')
2294     path.unlink()
2295 time_1 = time.time()
2296 print(f'Persisting {pdf_path} ...')
2297 result = abjad.persist(score_file).as_pdf(pdf_path)
2298 print(result[0])
2299 print(result[1])
2300 print(result[2])
2301 success = result[3]
2302 if success is False:
2303     print('LilyPond failed!')
2304 time_2 = time.time()
2305 total_time = time_2 - time_1
2306 print(f'Total time: {total_time} seconds')
2307 if path.exists():
2308     print(f'Opening {pdf_path} ...')
2309     os.system(f'open {pdf_path}')
2310 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_G/Section_G.
ly').readlines()
2311 open('/Users/evansdsg2/Scores/guerrero/Build/Section_G.ly', 'w').writelines(
    score_lines[15:-1])
2312
2313 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
2314     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2315     signature_copy = abjad.mutate(signatures).copy()
2316     staff_copy = abjad.mutate(staff).copy()
2317     part = abjad.Score()
2318     part.insert(0, staff)
2319     part.insert(0, signature_copy)
2320     part_file = abjad.LilyPondFile.new(
2321         part,
2322         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2323 _stylesheets/abjad.ily'],
2324     )
2325 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.) soprano'
2326 pdf_path = f'{directory}/Section_G.pdf'
2327 path = pathlib.Path('Section_G.pdf')
2328 if path.exists():
2329     print(f'Removing {pdf_path} ...')
2330     path.unlink()
2331 time_1 = time.time()
2332 print(f'Persisting {pdf_path} ...')
2333 result = abjad.persist(part_file).as_pdf(pdf_path)

```

```

2333 print(result[0])
2334 print(result[1])
2335 print(result[2])
2336 success = result[3]
2337 if success is False:
2338     print('LilyPond failed!')
2339 time_2 = time.time()
2340 total_time = time_2 - time_1
2341 print(f'Total time: {total_time} seconds')
2342 if path.exists():
2343     print(f'Opening {pdf_path} ...')
2344     os.system(f'open {pdf_path}')
2345 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano1/
Section_G.ly').readlines()
2346 open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.)soprano1/Section_G.ly',
'w').writelines(part_lines[15:-1])
2347
2348 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
2349     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2350     signature_copy = abjad.mutate(signatures).copy()
2351     staff_copy = abjad.mutate(staff).copy()
2352     part = abjad.Score()
2353     part.insert(0, staff)
2354     part.insert(0, signature_copy)
2355     part_file = abjad.LilyPondFile.new(
2356         part,
2357         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2358         )
2359     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1'
2360     pdf_path = f'{directory}/Section_G.pdf'
2361     path = pathlib.Path('Section_G.pdf')
2362     if path.exists():
2363         print(f'Removing {pdf_path} ...')
2364         path.unlink()
2365     time_1 = time.time()
2366     print(f'Persisting {pdf_path} ...')
2367     result = abjad.persist(part_file).as_pdf(pdf_path)
2368     print(result[0])
2369     print(result[1])
2370     print(result[2])
2371     success = result[3]
2372     if success is False:
2373         print('LilyPond failed!')
2374     time_2 = time.time()
2375     total_time = time_2 - time_1
2376     print(f'Total time: {total_time} seconds')
2377     if path.exists():
2378         print(f'Opening {pdf_path} ...')
2379         os.system(f'open {pdf_path}')
2380     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/
Section_G.ly').readlines()
2381     open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/Section_G.ly',
'w').writelines(part_lines[15:-1])
2382
2383 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
2384     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2385     signature_copy = abjad.mutate(signatures).copy()
2386     staff_copy = abjad.mutate(staff).copy()

```

```

2387 part = abjad.Score()
2388 part.insert(0, staff)
2389 part.insert(0, signature_copy)
2390 part_file = abjad.LilyPondFile.new(
2391     part,
2392     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2393 _stylesheets/abjad.ily'],
2394     )
2395 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
2396 pdf_path = f'{directory}/Section_G.pdf'
2397 path = pathlib.Path('Section_G.pdf')
2398 if path.exists():
2399     print(f'Removing {pdf_path} ...')
2400     path.unlink()
2401 time_1 = time.time()
2402 print(f'Persisting {pdf_path} ...')
2403 result = abjad.persist(part_file).as_pdf(pdf_path)
2404 print(result[0])
2405 print(result[1])
2406 print(result[2])
2407 success = result[3]
2408 if success is False:
2409     print('LilyPond failed!')
2410 time_2 = time.time()
2411 total_time = time_2 - time_1
2412 print(f'Total time: {total_time} seconds')
2413 if path.exists():
2414     print(f'Opening {pdf_path} ...')
2415     os.system(f'open {pdf_path}')
2416 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/
2417 Section_G.ly').readlines()
2418 open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/Section_G.ly',
2419 'w').writelines(part_lines[15:-1])
2420
2421 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
2422     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2423     signature_copy = abjad.mutate(signatures).copy()
2424     staff_copy = abjad.mutate(staff).copy()
2425     part = abjad.Score()
2426     part.insert(0, staff)
2427     part.insert(0, signature_copy)
2428     part_file = abjad.LilyPondFile.new(
2429         part,
2430         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2431 _stylesheets/abjad.ily'],
2432         )
2433 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3'
2434 pdf_path = f'{directory}/Section_G.pdf'
2435 path = pathlib.Path('Section_G.pdf')
2436 if path.exists():
2437     print(f'Removing {pdf_path} ...')
2438     path.unlink()
2439 time_1 = time.time()
2440 print(f'Persisting {pdf_path} ...')
2441 result = abjad.persist(part_file).as_pdf(pdf_path)
2442 print(result[0])
2443 print(result[1])
2444 print(result[2])
2445 success = result[3]
```

```

2442     if success is False:
2443         print('LilyPond failed!')
2444     time_2 = time.time()
2445     total_time = time_2 - time_1
2446     print(f'Total time: {total_time} seconds')
2447     if path.exists():
2448         print(f'Opening {pdf_path} ...')
2449         os.system(f'open {pdf_path}')
2450     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/
Section_G.ly').readlines()
2451     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/Section_G.ly',
'w').writelines(part_lines[15:-1])
2452
2453 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
2454     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2455     signature_copy = abjad.mutate(signatures).copy()
2456     staff_copy = abjad.mutate(staff).copy()
2457     part = abjad.Score()
2458     part.insert(0, staff)
2459     part.insert(0, signature_copy)
2460     part_file = abjad.LilyPondFile.new(
2461         part,
2462         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2463         )
2464     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1'
2465     pdf_path = f'{directory}/Section_G.pdf'
2466     path = pathlib.Path('Section_G.pdf')
2467     if path.exists():
2468         print(f'Removing {pdf_path} ...')
2469         path.unlink()
2470     time_1 = time.time()
2471     print(f'Persisting {pdf_path} ...')
2472     result = abjad.persist(part_file).as_pdf(pdf_path)
2473     print(result[0])
2474     print(result[1])
2475     print(result[2])
2476     success = result[3]
2477     if success is False:
2478         print('LilyPond failed!')
2479     time_2 = time.time()
2480     total_time = time_2 - time_1
2481     print(f'Total time: {total_time} seconds')
2482     if path.exists():
2483         print(f'Opening {pdf_path} ...')
2484         os.system(f'open {pdf_path}')
2485     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/
Section_G.ly').readlines()
2486     open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/Section_G.ly',
'w').writelines(part_lines[15:-1])
2487
2488 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
2489     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2490     signature_copy = abjad.mutate(signatures).copy()
2491     staff_copy = abjad.mutate(staff).copy()
2492     part = abjad.Score()
2493     part.insert(0, staff)
2494     part.insert(0, signature_copy)
2495     part_file = abjad.LilyPondFile.new(

```

```

2496     part,
2497     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2498 _stylesheets/abjad.ily'],
2499     )
2500 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
2501 pdf_path = f'{directory}/Section_G.pdf'
2502 path = pathlib.Path('Section_G.pdf')
2503 if path.exists():
2504     print(f'Removing {pdf_path} ...')
2505     path.unlink()
2506 time_1 = time.time()
2507 print(f'Persisting {pdf_path} ...')
2508 result = abjad.persist(part_file).as_pdf(pdf_path)
2509 print(result[0])
2510 print(result[1])
2511 print(result[2])
2512 success = result[3]
2513 if success is False:
2514     print('LilyPond failed!')
2515 time_2 = time.time()
2516 total_time = time_2 - time_1
2517 print(f'Total time: {total_time} seconds')
2518 if path.exists():
2519     print(f'Opening {pdf_path} ...')
2520     os.system(f'open {pdf_path}')
2521 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
2522 Section_G.ly').readlines()
2523 open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_G.ly', 'w'
2524 ).writelines(part_lines[15:-1])
2525
2526 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
2527     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2528     signature_copy = abjad.mutate(signatures).copy()
2529     staff_copy = abjad.mutate(staff).copy()
2530     part = abjad.Score()
2531     part.insert(0, staff)
2532     part.insert(0, signature_copy)
2533     part_file = abjad.LilyPondFile.new(
2534         part,
2535         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2536 _stylesheets/abjad.ily'],
2537         )
2538 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
2539 pdf_path = f'{directory}/Section_G.pdf'
2540 path = pathlib.Path('Section_G.pdf')
2541 if path.exists():
2542     print(f'Removing {pdf_path} ...')
2543     path.unlink()
2544 time_1 = time.time()
2545 print(f'Persisting {pdf_path} ...')
2546 result = abjad.persist(part_file).as_pdf(pdf_path)
2547 print(result[0])
2548 print(result[1])
2549 print(result[2])
2550 success = result[3]
2551 if success is False:
2552     print('LilyPond failed!')
2553 time_2 = time.time()
2554 total_time = time_2 - time_1

```

```

2551 print(f'Total time: {total_time} seconds')
2552 if path.exists():
2553     print(f'Opening {pdf_path} ...')
2554     os.system(f'open {pdf_path}')
2555 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/
Section_G.ly').readlines()
2556 open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/Section_G.ly', 'w'
).writelines(part_lines[15:-1])
2557
2558 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
2559     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2560     signature_copy = abjad.mutate(signatures).copy()
2561     staff_copy = abjad.mutate(staff).copy()
2562     part = abjad.Score()
2563     part.insert(0, staff)
2564     part.insert(0, signature_copy)
2565     part_file = abjad.LilyPondFile.new(
2566         part,
2567         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2568         )
2569     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
2570     pdf_path = f'{directory}/Section_G.pdf'
2571     path = pathlib.Path('Section_G.pdf')
2572     if path.exists():
2573         print(f'Removing {pdf_path} ...')
2574         path.unlink()
2575     time_1 = time.time()
2576     print(f'Persisting {pdf_path} ...')
2577     result = abjad.persist(part_file).as_pdf(pdf_path)
2578     print(result[0])
2579     print(result[1])
2580     print(result[2])
2581     success = result[3]
2582     if success is False:
2583         print('LilyPond failed!')
2584     time_2 = time.time()
2585     total_time = time_2 - time_1
2586     print(f'Total time: {total_time} seconds')
2587     if path.exists():
2588         print(f'Opening {pdf_path} ...')
2589         os.system(f'open {pdf_path}')
2590     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/
Section_G.ly').readlines()
2591     open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/Section_G.ly', 'w'
).writelines(part_lines[15:-1])
2592
2593 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
2594     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2595     signature_copy = abjad.mutate(signatures).copy()
2596     staff_copy = abjad.mutate(staff).copy()
2597     part = abjad.Score()
2598     part.insert(0, staff)
2599     part.insert(0, signature_copy)
2600     part_file = abjad.LilyPondFile.new(
2601         part,
2602         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2603         )

```

```

2604 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
2605 pdf_path = f'{directory}/Section_G.pdf'
2606 path = pathlib.Path('Section_G.pdf')
2607 if path.exists():
2608     print(f'Removing {pdf_path} ...')
2609     path.unlink()
2610 time_1 = time.time()
2611 print(f'Persisting {pdf_path} ...')
2612 result = abjad.persist(part_file).as_pdf(pdf_path)
2613 print(result[0])
2614 print(result[1])
2615 print(result[2])
2616 success = result[3]
2617 if success is False:
2618     print('LilyPond failed!')
2619 time_2 = time.time()
2620 total_time = time_2 - time_1
2621 print(f'Total time: {total_time} seconds')
2622 if path.exists():
2623     print(f'Opening {pdf_path} ...')
2624     os.system(f'open {pdf_path}')
2625 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/
2626 Section_G.ly').readlines()
2627 open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/Section_G.ly', 'w'
2628 ).writelines(part_lines[15:-1])
2629
2630 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
2631     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2632     signature_copy = abjad.mutate(signatures).copy()
2633     staff_copy = abjad.mutate(staff).copy()
2634     part = abjad.Score()
2635     part.insert(0, staff)
2636     part.insert(0, signature_copy)
2637     part_file = abjad.LilyPondFile.new(
2638         part,
2639         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2640         _stylesheets/abjad.ily'],
2641         )
2642     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
2643     pdf_path = f'{directory}/Section_G.pdf'
2644     path = pathlib.Path('Section_G.pdf')
2645     if path.exists():
2646         print(f'Removing {pdf_path} ...')
2647         path.unlink()
2648     time_1 = time.time()
2649     print(f'Persisting {pdf_path} ...')
2650     result = abjad.persist(part_file).as_pdf(pdf_path)
2651     print(result[0])
2652     print(result[1])
2653     print(result[2])
2654     success = result[3]
2655     if success is False:
2656         print('LilyPond failed!')
2657     time_2 = time.time()
2658     total_time = time_2 - time_1
2659     print(f'Total time: {total_time} seconds')
2660     if path.exists():
2661         print(f'Opening {pdf_path} ...')
2662         os.system(f'open {pdf_path}')

```

```

2660 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
2661 Section_G.ly').readlines()
2662 open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_G.ly', 'w'
2663 ).writelines(part_lines[15:-1])
2664
2665 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
2666     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2667     signature_copy = abjad.mutate(signatures).copy()
2668     staff_copy = abjad.mutate(staff).copy()
2669     part = abjad.Score()
2670     part.insert(0, staff)
2671     part.insert(0, signature_copy)
2672     part_file = abjad.LilyPondFile.new(
2673         part,
2674         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2675 _stylesheets/abjad.ily'],
2676         )
2677     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
2678     pdf_path = f'{directory}/Section_G.pdf'
2679     path = pathlib.Path('Section_G.pdf')
2680     if path.exists():
2681         print(f'Removing {pdf_path} ...')
2682         path.unlink()
2683     time_1 = time.time()
2684     print(f'Persisting {pdf_path} ...')
2685     result = abjad.persist(part_file).as_pdf(pdf_path)
2686     print(result[0])
2687     print(result[1])
2688     print(result[2])
2689     success = result[3]
2690     if success is False:
2691         print('LilyPond failed!')
2692     time_2 = time.time()
2693     total_time = time_2 - time_1
2694     print(f'Total time: {total_time} seconds')
2695     if path.exists():
2696         print(f'Opening {pdf_path} ...')
2697         os.system(f'open {pdf_path}')
2698     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
2699 Section_G.ly').readlines()
2700     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Section_G.ly', 'w'
2701 ).writelines(part_lines[15:-1])
2702
2703 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
2704     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2705     signature_copy = abjad.mutate(signatures).copy()
2706     staff_copy = abjad.mutate(staff).copy()
2707     part = abjad.Score()
2708     part.insert(0, staff)
2709     part.insert(0, signature_copy)
2710     part_file = abjad.LilyPondFile.new(
2711         part,
2712         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2713 _stylesheets/abjad.ily'],
2714         )
2715     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
2716     pdf_path = f'{directory}/Section_G.pdf'
2717     path = pathlib.Path('Section_G.pdf')
2718     if path.exists():

```

```

2713     print(f'Removing {pdf_path} ...')
2714     path.unlink()
2715 time_1 = time.time()
2716 print(f'Persisting {pdf_path} ...')
2717 result = abjad.persist(part_file).as_pdf(pdf_path)
2718 print(result[0])
2719 print(result[1])
2720 print(result[2])
2721 success = result[3]
2722 if success is False:
2723     print('LilyPond failed!')
2724 time_2 = time.time()
2725 total_time = time_2 - time_1
2726 print(f'Total time: {total_time} seconds')
2727 if path.exists():
2728     print(f'Opening {pdf_path} ...')
2729     os.system(f'open {pdf_path}')
2730 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/
Section_G.ly').readlines()
2731 open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.')tenor2/Section_G.ly', ,
w').writelines(part_lines[15:-1])
2732
2733 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
2734     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2735     signature_copy = abjad.mutate(signatures).copy()
2736     staff_copy = abjad.mutate(staff).copy()
2737     part = abjad.Score()
2738     part.insert(0, staff)
2739     part.insert(0, signature_copy)
2740     part_file = abjad.LilyPondFile.new(
2741         part,
2742         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2743         )
2744     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3'
2745     pdf_path = f'{directory}/Section_G.pdf'
2746     path = pathlib.Path('Section_G.pdf')
2747     if path.exists():
2748         print(f'Removing {pdf_path} ...')
2749         path.unlink()
2750     time_1 = time.time()
2751     print(f'Persisting {pdf_path} ...')
2752     result = abjad.persist(part_file).as_pdf(pdf_path)
2753     print(result[0])
2754     print(result[1])
2755     print(result[2])
2756     success = result[3]
2757     if success is False:
2758         print('LilyPond failed!')
2759     time_2 = time.time()
2760     total_time = time_2 - time_1
2761     print(f'Total time: {total_time} seconds')
2762     if path.exists():
2763         print(f'Opening {pdf_path} ...')
2764         os.system(f'open {pdf_path}')
2765     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
Section_G.ly').readlines()
2766     open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_G.ly', ,
w').writelines(part_lines[15:-1])

```

```

2767
2768 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
2769     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2770     signature_copy = abjad.mutate(signatures).copy()
2771     staff_copy = abjad.mutate(staff).copy()
2772     part = abjad.Score()
2773     part.insert(0, staff)
2774     part.insert(0, signature_copy)
2775     part_file = abjad.LilyPondFile.new(
2776         part,
2777         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2778 _stylesheets/abjad.ily'],
2779         )
2780     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.)tenor4'
2781     pdf_path = f'{directory}/Section_G.pdf'
2782     path = pathlib.Path('Section_G.pdf')
2783     if path.exists():
2784         print(f'Removing {pdf_path} ...')
2785         path.unlink()
2786     time_1 = time.time()
2787     print(f'Persisting {pdf_path} ...')
2788     result = abjad.persist(part_file).as_pdf(pdf_path)
2789     print(result[0])
2790     print(result[1])
2791     print(result[2])
2792     success = result[3]
2793     if success is False:
2794         print('LilyPond failed!')
2795     time_2 = time.time()
2796     total_time = time_2 - time_1
2797     print(f'Total time: {total_time} seconds')
2798     if path.exists():
2799         print(f'Opening {pdf_path} ...')
2800         os.system(f'open {pdf_path}')
2801     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.)tenor4/
2802 Section_G.ly').readlines()
2803     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.)tenor4/Section_G.ly', 'w').writelines(part_lines[15:-1])
2804
2805 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
2806     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2807     signature_copy = abjad.mutate(signatures).copy()
2808     staff_copy = abjad.mutate(staff).copy()
2809     part = abjad.Score()
2810     part.insert(0, staff)
2811     part.insert(0, signature_copy)
2812     part_file = abjad.LilyPondFile.new(
2813         part,
2814         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2815 _stylesheets/abjad.ily'],
2816         )
2817     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5'
2818     pdf_path = f'{directory}/Section_G.pdf'
2819     path = pathlib.Path('Section_G.pdf')
2820     if path.exists():
2821         print(f'Removing {pdf_path} ...')

```

```

2822     result = abjad.persist(part_file).as_pdf(pdf_path)
2823     print(result[0])
2824     print(result[1])
2825     print(result[2])
2826     success = result[3]
2827     if success is False:
2828         print('LilyPond failed!')
2829     time_2 = time.time()
2830     total_time = time_2 - time_1
2831     print(f'Total time: {total_time} seconds')
2832     if path.exists():
2833         print(f'Opening {pdf_path} ...')
2834         os.system(f'open {pdf_path}')
2835     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/
2836     Section_G.ly').readlines()
2837     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/Section_G.ly', ,
2838     'w').writelines(part_lines[15:-1])
2839
2840 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
2841     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2842     signature_copy = abjad.mutate(signatures).copy()
2843     staff_copy = abjad.mutate(staff).copy()
2844     part = abjad.Score()
2845     part.insert(0, staff)
2846     part.insert(0, signature_copy)
2847     part_file = abjad.LilyPondFile.new(
2848         part,
2849         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2850         _stylesheets/abjad.ily'],
2851         )
2852     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1'
2853     pdf_path = f'{directory}/Section_G.pdf'
2854     path = pathlib.Path('Section_G.pdf')
2855     if path.exists():
2856         print(f'Removing {pdf_path} ...')
2857         path.unlink()
2858     time_1 = time.time()
2859     print(f'Persisting {pdf_path} ...')
2860     result = abjad.persist(part_file).as_pdf(pdf_path)
2861     print(result[0])
2862     print(result[1])
2863     print(result[2])
2864     success = result[3]
2865     if success is False:
2866         print('LilyPond failed!')
2867     time_2 = time.time()
2868     total_time = time_2 - time_1
2869     print(f'Total time: {total_time} seconds')
2870     if path.exists():
2871         print(f'Opening {pdf_path} ...')
2872         os.system(f'open {pdf_path}')
2873     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/
2874     Section_G.ly').readlines()
2875     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/Section_G.ly',
2876     'w').writelines(part_lines[15:-1])
2877
2878 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
2879     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2880     signature_copy = abjad.mutate(signatures).copy()

```

```

2876 staff_copy = abjad.mutate(staff).copy()
2877 part = abjad.Score()
2878 part.insert(0, staff)
2879 part.insert(0, signature_copy)
2880 part_file = abjad.LilyPondFile.new(
2881     part,
2882     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2883 _stylesheets/abjad.ily'],
2884     )
2885 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2'
2886 pdf_path = f'{directory}/Section_G.pdf'
2887 path = pathlib.Path('Section_G.pdf')
2888 if path.exists():
2889     print(f'Removing {pdf_path} ...')
2890     path.unlink()
2891 time_1 = time.time()
2892 print(f'Persisting {pdf_path} ...')
2893 result = abjad.persist(part_file).as_pdf(pdf_path)
2894 print(result[0])
2895 print(result[1])
2896 print(result[2])
2897 success = result[3]
2898 if success is False:
2899     print('LilyPond failed!')
2900 time_2 = time.time()
2901 total_time = time_2 - time_1
2902 print(f'Total time: {total_time} seconds')
2903 if path.exists():
2904     print(f'Opening {pdf_path} ...')
2905     os.system(f'open {pdf_path}')
2906 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/
2907 Section_G.ly').readlines()
2908 open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/Section_G.ly',
2909 'w').writelines(part_lines[15:-1])
2910
2911 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
2912     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2913     signature_copy = abjad.mutate(signatures).copy()
2914     staff_copy = abjad.mutate(staff).copy()
2915     part = abjad.Score()
2916     part.insert(0, staff)
2917     part.insert(0, signature_copy)
2918     part_file = abjad.LilyPondFile.new(
2919         part,
2920         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2921 _stylesheets/abjad.ily'],
2922         )
2923     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3'
2924     pdf_path = f'{directory}/Section_G.pdf'
2925     path = pathlib.Path('Section_G.pdf')
2926     if path.exists():
2927         print(f'Removing {pdf_path} ...')
2928         path.unlink()
2929     time_1 = time.time()
2930     print(f'Persisting {pdf_path} ...')
2931     result = abjad.persist(part_file).as_pdf(pdf_path)
2932     print(result[0])
2933     print(result[1])
2934     print(result[2])

```

```

2931     success = result[3]
2932     if success is False:
2933         print('LilyPond failed!')
2934     time_2 = time.time()
2935     total_time = time_2 - time_1
2936     print(f'Total time: {total_time} seconds')
2937     if path.exists():
2938         print(f'Opening {pdf_path} ...')
2939         os.system(f'open {pdf_path}')
2940     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/
2941     Section_G.ly').readlines()
2942     open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/Section_G.ly',
2943           'w').writelines(part_lines[15:-1])
2944
2945 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
2946     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2947     signature_copy = abjad.mutate(signatures).copy()
2948     staff_copy = abjad.mutate(staff).copy()
2949     part = abjad.Score()
2950     part.insert(0, staff)
2951     part.insert(0, signature_copy)
2952     part_file = abjad.LilyPondFile.new(
2953         part,
2954         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2955         _stylesheets/abjad.ily'],
2956         )
2957     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
2958     pdf_path = f'{directory}/Section_G.pdf'
2959     path = pathlib.Path('Section_G.pdf')
2960     if path.exists():
2961         print(f'Removing {pdf_path} ...')
2962         path.unlink()
2963     time_1 = time.time()
2964     print(f'Persisting {pdf_path} ...')
2965     result = abjad.persist(part_file).as_pdf(pdf_path)
2966     print(result[0])
2967     print(result[1])
2968     print(result[2])
2969     success = result[3]
2970     if success is False:
2971         print('LilyPond failed!')
2972     time_2 = time.time()
2973     total_time = time_2 - time_1
2974     print(f'Total time: {total_time} seconds')
2975     if path.exists():
2976         print(f'Opening {pdf_path} ...')
2977         os.system(f'open {pdf_path}')
2978     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
2979     Section_G.ly').readlines()
2980     open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_G.ly',
2981           'w').writelines(part_lines[15:-1])
2982
2983 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
2984     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2985     signature_copy = abjad.mutate(signatures).copy()
2986     staff_copy = abjad.mutate(staff).copy()
2987     part = abjad.Score()
2988     part.insert(0, staff)
2989     part.insert(0, signature_copy)

```

```

2985 part_file = abjad.LilyPondFile.new(
2986     part,
2987     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2988 _stylesheets/abjad.ily'],
2989     )
2990 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
2991 pdf_path = f'{directory}/Section_G.pdf'
2992 path = pathlib.Path('Section_G.pdf')
2993 if path.exists():
2994     print(f'Removing {pdf_path} ...')
2995     path.unlink()
2996 time_1 = time.time()
2997 print(f'Persisting {pdf_path} ...')
2998 result = abjad.persist(part_file).as_pdf(pdf_path)
2999 print(result[0])
3000 print(result[1])
3001 print(result[2])
3002 success = result[3]
3003 if success is False:
3004     print('LilyPond failed!')
3005 time_2 = time.time()
3006 total_time = time_2 - time_1
3007 print(f'Total time: {total_time} seconds')
3008 if path.exists():
3009     print(f'Opening {pdf_path} ...')
3010     os.system(f'open {pdf_path}')
3011 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
3012 Section_G.ly').readlines()
3013 open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_G.ly', 'w'
3014 .writelines(part_lines[15:-1])
3015
3016 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
3017     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3018     signature_copy = abjad.mutate(signatures).copy()
3019     staff_copy = abjad.mutate(staff).copy()
3020     part = abjad.Score()
3021     part.insert(0, staff)
3022     part.insert(0, signature_copy)
3023     part_file = abjad.LilyPondFile.new(
3024         part,
3025         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3026 _stylesheets/abjad.ily'],
3027         )
3028 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
3029 pdf_path = f'{directory}/Section_G.pdf'
3030 path = pathlib.Path('Section_G.pdf')
3031 if path.exists():
3032     print(f'Removing {pdf_path} ...')
3033     path.unlink()
3034 time_1 = time.time()
3035 print(f'Persisting {pdf_path} ...')
3036 result = abjad.persist(part_file).as_pdf(pdf_path)
3037 print(result[0])
3038 print(result[1])
3039 print(result[2])
3040 success = result[3]
3041 if success is False:
3042     print('LilyPond failed!')
3043 time_2 = time.time()

```

```

3040     total_time = time_2 - time_1
3041     print(f'Total time: {total_time} seconds')
3042     if path.exists():
3043         print(f'Opening {pdf_path} ...')
3044         os.system(f'open {pdf_path}')
3045     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.')contrabass
3046     /Section_G.ly').readlines()
3047     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.')contrabass/Section_G.ly
3048     ', 'w').writelines(part_lines[15:-1])

```

Listing 3.8: Invocation Source Code

## 3.9 Section H

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11
12 print('Interpreting file ...')
13
14 time_signatures = [
15     abjad.TimeSignature(pair) for pair in [
16         (5, 4), (4, 4), (3, 4), (5, 4), (4, 4), (3, 4),
17         (3, 4), (4, 4), (5, 4), (3, 4), (4, 4), (5, 4),
18     ]
19 ]
20
21 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
22
23 def cyc(lst):
24     count = 0
25     while True:
26         yield lst[count%len(lst)]
27         count += 1
28
29 def grouper(lst1, lst2):
30     def cyc(lst):
31         c = 0
32         while True:
33             yield lst[c%len(lst)]
34             c += 1
35     lst1 = cyc(lst1)
36     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2
37     ]
38
39 soprano_notes = [14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20,
40     20.5, 21, 21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25, 25.5]
41 soprano_1_notes = [15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21,
42     21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25, 25.5, 26, 26.5]
43 soprano_2_notes = [11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17,

```

```

    17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5, 23]
41 soprano_3_notes = [11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17,
    17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5]
42 alto_1_notes = [14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20,
    20.5, 21, 21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25, 25.5]
43 alto_2_notes = [10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5, 15, 15.5, 16,
    16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5]
44 alto_3_notes = [6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13,
    13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5]
45 alto_4_notes = [1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9,
    9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13]
46 alto_5_notes = [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9,
    9.5, 10, 10.5, 11, 11.5, 12, 12.5]
47 alto_6_notes = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8,
    8.5, 9, 9.5, 10, 10.5, 11, 11.5]
48 tenor_1_notes = [7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5,
    14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5]
49 tenor_2_notes = [3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5,
    11, 11.5, 12, 12.5, 13, 13.5, 14, 14.5]
50 tenor_3_notes = [-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,
    7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5]
51 tenor_4_notes = [0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8,
    8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12]
52 tenor_5_notes = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5,
    8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5]
53 baritone_1_notes = [2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9,
    9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13, 13.5]
54 baritone_2_notes = [-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5,
    5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5]
55 baritone_3_notes = [-1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5,
    5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10]
56 bass_1_notes = [-0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7,
    7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11]
57 bass_2_notes = [-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,
    7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5]
58 contrabass_notes = [6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5,
    13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5]
59
60 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
61     talea=abjadext.rmakers.Talea(
62         counts=[1, 1, 1, 5, 3, 2, 4],
63         denominator=8,
64     ),
65     beamSpecifier=abjadext.rmakers.BeamSpecifier(
66         beamDivisionsTogether=True,
67         beamRests=False,
68     ),
69     extraCountsPerDivision=[0, 1, -1, ],
70     logicalTieMasks=[
71         abjadext.rmakers.silence([8], 5),
72     ],
73     divisionMasks=[
74         abjadext.rmakers.SilenceMask(
75             pattern=abjad.index([2], 11),
76         ),
77     ],
78     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
79         trivialize=True,
80         extractTrivial=True,

```

```

81     rewrite_rest_filled=True,
82     rewrite_sustained=True,
83   ),
84 )
85
86 rmaker_two = abjadext.rmakers.TaleaRhythmMaker(
87   talea=abjadext.rmakers.Talea(
88     counts=[4, 3, 1, 5, 2],
89     denominator=8,
90   ),
91   beamSpecifier=abjadext.rmakers.BeamSpecifier(
92     beamDivisionsTogether=True,
93     beamRests=False,
94   ),
95   extraCountsPerDivision=[-1, 0, -1, 1, 0, ],
96   logicalTieMasks=[
97     abjadext.rmakers.silence([8], 5),
98   ],
99   divisionMasks=[
100     abjadext.rmakers.SilenceMask(
101       pattern=abjad.index([2], 11),
102     ),
103   ],
104   tupletSpecifier=abjadext.rmakers.TupletSpecifier(
105     trivialize=True,
106     extractTrivial=True,
107     rewriteRestFilled=True,
108     rewriteSustained=True,
109   ),
110 )
111
112 attachmentHandlerOne = AttachmentHandler(
113   startingDynamic='mp',
114   endingDynamic='mf',
115   hairpin='<',
116   articulationList=['halfopen', '', 'stopped', 'halfopen', 'stopped', ],
117 )
118
119 attachmentHandlerTwo = AttachmentHandler(
120   startingDynamic='mf',
121   endingDynamic='p',
122   hairpin='>',
123   articulationList=['flageolet', 'flageolet', '', 'flageolet', 'stopped', ],
124 )
125
126 #####sopranino#####
127 sopraninoMusicmakerOne = MusicMaker(
128   rmaker=rmakerOne,
129   pitches=sopraninoNotes,
130   continuous=True,
131   attachmentHandler=attachmentHandlerOne,
132 )
133 sopraninoMusicmakerTwo = MusicMaker(
134   rmaker=rmakerTwo,
135   pitches=sopraninoNotes,
136   continuous=True,
137   attachmentHandler=attachmentHandlerTwo,
138 )
139 #####soprano_one#####

```

```
140 soprano_one_musicmaker_one = MusicMaker(
141     rmaker=rmaker_one,
142     pitches=soprano_1_notes,
143     continuous=True,
144     attachment_handler=attachment_handler_one,
145 )
146 soprano_one_musicmaker_two = MusicMaker(
147     rmaker=rmaker_two,
148     pitches=soprano_1_notes,
149     continuous=True,
150     attachment_handler=attachment_handler_two,
151 )
152 #####soprano_two#####
153 soprano_two_musicmaker_one = MusicMaker(
154     rmaker=rmaker_one,
155     pitches=soprano_2_notes,
156     continuous=True,
157     attachment_handler=attachment_handler_one,
158 )
159 soprano_two_musicmaker_two = MusicMaker(
160     rmaker=rmaker_two,
161     pitches=soprano_2_notes,
162     continuous=True,
163     attachment_handler=attachment_handler_two,
164 )
165 #####soprano_three#####
166 soprano_three_musicmaker_one = MusicMaker(
167     rmaker=rmaker_one,
168     pitches=soprano_3_notes,
169     continuous=True,
170     attachment_handler=attachment_handler_one,
171 )
172 soprano_three_musicmaker_two = MusicMaker(
173     rmaker=rmaker_two,
174     pitches=soprano_3_notes,
175     continuous=True,
176     attachment_handler=attachment_handler_two,
177 )
178 #####alto_one#####
179 alto_one_musicmaker_one = MusicMaker(
180     rmaker=rmaker_one,
181     pitches=alto_1_notes,
182     continuous=True,
183     attachment_handler=attachment_handler_one,
184 )
185 alto_one_musicmaker_two = MusicMaker(
186     rmaker=rmaker_two,
187     pitches=alto_1_notes,
188     continuous=True,
189     attachment_handler=attachment_handler_two,
190 )
191 #####alto_two#####
192 alto_two_musicmaker_one = MusicMaker(
193     rmaker=rmaker_one,
194     pitches=alto_2_notes,
195     continuous=True,
196     attachment_handler=attachment_handler_one,
197 )
198 alto_two_musicmaker_two = MusicMaker(
```

```
199     rmaker=rmaker_two ,
200     pitches=alto_2_notes ,
201     continuous=True ,
202     attachment_handler=attachment_handler_two ,
203 )
204 #####alto_three#####
205 alto_three_musicmaker_one = MusicMaker(
206     rmaker=rmaker_one ,
207     pitches=alto_3_notes ,
208     continuous=True ,
209     attachment_handler=attachment_handler_one ,
210 )
211 alto_three_musicmaker_two = MusicMaker(
212     rmaker=rmaker_two ,
213     pitches=alto_3_notes ,
214     continuous=True ,
215     attachment_handler=attachment_handler_two ,
216 )
217 #####alto_four#####
218 alto_four_musicmaker_one = MusicMaker(
219     rmaker=rmaker_one ,
220     pitches=alto_4_notes ,
221     continuous=True ,
222     attachment_handler=attachment_handler_one ,
223 )
224 alto_four_musicmaker_two = MusicMaker(
225     rmaker=rmaker_two ,
226     pitches=alto_4_notes ,
227     continuous=True ,
228     attachment_handler=attachment_handler_two ,
229 )
230 #####alto_five#####
231 alto_five_musicmaker_one = MusicMaker(
232     rmaker=rmaker_one ,
233     pitches=alto_5_notes ,
234     continuous=True ,
235     attachment_handler=attachment_handler_one ,
236 )
237 alto_five_musicmaker_two = MusicMaker(
238     rmaker=rmaker_two ,
239     pitches=alto_5_notes ,
240     continuous=True ,
241     attachment_handler=attachment_handler_two ,
242 )
243 #####alto_six#####
244 alto_six_musicmaker_one = MusicMaker(
245     rmaker=rmaker_one ,
246     pitches=alto_6_notes ,
247     continuous=True ,
248     attachment_handler=attachment_handler_one ,
249 )
250 alto_six_musicmaker_two = MusicMaker(
251     rmaker=rmaker_two ,
252     pitches=alto_6_notes ,
253     continuous=True ,
254     attachment_handler=attachment_handler_two ,
255 )
256 #####tenor_one#####
257 tenor_one_musicmaker_one = MusicMaker(
```

```
258     rmaker=rmaker_one,
259     pitches=tenor_1_notes,
260     continuous=True,
261     attachment_handler=attachment_handler_one,
262 )
263 tenor_one_musicmaker_two = MusicMaker(
264     rmaker=rmaker_two,
265     pitches=tenor_1_notes,
266     continuous=True,
267     attachment_handler=attachment_handler_two,
268 )
269 #####tenor_two#####
270 tenor_two_musicmaker_one = MusicMaker(
271     rmaker=rmaker_one,
272     pitches=tenor_2_notes,
273     continuous=True,
274     attachment_handler=attachment_handler_one,
275 )
276 tenor_two_musicmaker_two = MusicMaker(
277     rmaker=rmaker_two,
278     pitches=tenor_2_notes,
279     continuous=True,
280     attachment_handler=attachment_handler_two,
281 )
282 #####tenor_three#####
283 tenor_three_musicmaker_one = MusicMaker(
284     rmaker=rmaker_one,
285     pitches=tenor_3_notes,
286     continuous=True,
287     attachment_handler=attachment_handler_one,
288 )
289 tenor_three_musicmaker_two = MusicMaker(
290     rmaker=rmaker_two,
291     pitches=tenor_3_notes,
292     continuous=True,
293     attachment_handler=attachment_handler_two,
294 )
295 #####tenor_four#####
296 tenor_four_musicmaker_one = MusicMaker(
297     rmaker=rmaker_one,
298     pitches=tenor_4_notes,
299     continuous=True,
300     attachment_handler=attachment_handler_one,
301 )
302 tenor_four_musicmaker_two = MusicMaker(
303     rmaker=rmaker_two,
304     pitches=tenor_4_notes,
305     continuous=True,
306     attachment_handler=attachment_handler_two,
307 )
308 #####tenor_five#####
309 tenor_five_musicmaker_one = MusicMaker(
310     rmaker=rmaker_one,
311     pitches=tenor_5_notes,
312     continuous=True,
313     attachment_handler=attachment_handler_one,
314 )
315 tenor_five_musicmaker_two = MusicMaker(
316     rmaker=rmaker_two,
```

```
317     pitches=tenor_5_notes,
318     continuous=True,
319     attachment_handler=attachment_handler_two,
320 )
321 #####baritone_one#####
322 baritone_one_musicmaker_one = MusicMaker(
323     rmaker=rmaker_one,
324     pitches=baritone_1_notes,
325     continuous=True,
326     attachment_handler=attachment_handler_one,
327 )
328 baritone_one_musicmaker_two = MusicMaker(
329     rmaker=rmaker_two,
330     pitches=baritone_1_notes,
331     continuous=True,
332     attachment_handler=attachment_handler_two,
333 )
334 #####baritone_two#####
335 baritone_two_musicmaker_one = MusicMaker(
336     rmaker=rmaker_one,
337     pitches=baritone_2_notes,
338     continuous=True,
339     attachment_handler=attachment_handler_one,
340 )
341 baritone_two_musicmaker_two = MusicMaker(
342     rmaker=rmaker_two,
343     pitches=baritone_2_notes,
344     continuous=True,
345     attachment_handler=attachment_handler_two,
346 )
347 #####baritone_three#####
348 baritone_three_musicmaker_one = MusicMaker(
349     rmaker=rmaker_one,
350     pitches=baritone_3_notes,
351     continuous=True,
352     attachment_handler=attachment_handler_one,
353 )
354 baritone_three_musicmaker_two = MusicMaker(
355     rmaker=rmaker_two,
356     pitches=baritone_3_notes,
357     continuous=True,
358     attachment_handler=attachment_handler_two,
359 )
360 #####bass_one#####
361 bass_one_musicmaker_one = MusicMaker(
362     rmaker=rmaker_one,
363     pitches=bass_1_notes,
364     continuous=True,
365     attachment_handler=attachment_handler_one,
366 )
367 bass_one_musicmaker_two = MusicMaker(
368     rmaker=rmaker_two,
369     pitches=bass_1_notes,
370     continuous=True,
371     attachment_handler=attachment_handler_two,
372 )
373 #####bass_two#####
374 bass_two_musicmaker_one = MusicMaker(
375     rmaker=rmaker_one,
```

```

376     pitches=bass_2_notes ,
377     continuous=True ,
378     attachment_handler=attachment_handler_one ,
379 )
380 bass_two_musicmaker_two = MusicMaker(
381     rmaker=rmaker_two ,
382     pitches=bass_2_notes ,
383     continuous=True ,
384     attachment_handler=attachment_handler_two ,
385 )
386 #####contrabass#####
387 contrabass_musicmaker_one = MusicMaker(
388     rmaker=rmaker_one ,
389     pitches=contrabass_notes ,
390     continuous=True ,
391     attachment_handler=attachment_handler_one ,
392 )
393 contrabass_musicmaker_two = MusicMaker(
394     rmaker=rmaker_two ,
395     pitches=contrabass_notes ,
396     continuous=True ,
397     attachment_handler=attachment_handler_two ,
398 )
399
400 silence_maker = abjadext.rmakers.NoteRhythmMaker(
401     division_masks=[
402         abjadext.rmakers.SilenceMask(
403             pattern=abjad.index([0], 1),
404             ),
405         ],
406     )
407
408 class MusicSpecifier:
409
410     def __init__(self, music_maker, voice_name):
411         self.music_maker = music_maker
412         self.voice_name = voice_name
413
414 print('Collecting timespans and rmakers ...')
415
416 voice_1_timespan_list = abjad.TimespanList([
417     abjad.AnnotatedTimespan(
418         start_offset=start_offset,
419         stop_offset=stop_offset,
420         annotation=MusicSpecifier(
421             music_maker=music_maker,
422             voice_name='Voice 1',
423             ),
424     )
425     for start_offset, stop_offset, music_maker in [
426         [(0, 4), (2, 4), sopranino_musicmaker_one],
427         [(2, 4), (3, 4), sopranino_musicmaker_one],
428         [(5, 4), (7, 4), sopranino_musicmaker_one],
429         [(7, 4), (8, 4), sopranino_musicmaker_one],
430         [(12, 4), (14, 4), sopranino_musicmaker_two],
431         [(14, 4), (15, 4), sopranino_musicmaker_two],
432         [(17, 4), (18, 4), sopranino_musicmaker_one],
433         [(18, 4), (20, 4), sopranino_musicmaker_one],
434         [(28, 4), (31, 4), sopranino_musicmaker_two],

```

```

435     [(33, 4), (35, 4), soprano_musicmaker_two],
436     [(35, 4), (36, 4), soprano_musicmaker_two],
437     [(40, 4), (42, 4), soprano_musicmaker_one],
438     [(42, 4), (43, 4), soprano_musicmaker_one],
439     [(45, 4), (46, 4), soprano_musicmaker_two],
440     [(46, 4), (47, 4), soprano_musicmaker_two],
441     [(47, 4), (95, 8), soprano_musicmaker_two],
442     [(95, 8), (96, 8), silence_maker],
443 ]
444 ])
445
446 voice_2_timespan_list = abjad.TimespanList([
447     abjad.AnnotatedTimespan(
448         start_offset=start_offset,
449         stop_offset=stop_offset,
450         annotation=MusicSpecifier(
451             music_maker=music_maker,
452             voice_name='Voice 2',
453         ),
454     )
455     for start_offset, stop_offset, music_maker in [
456         [(4, 4), (5, 4), soprano_one_musicmaker_two],
457         [(5, 4), (6, 4), soprano_one_musicmaker_two],
458         [(6, 4), (7, 4), soprano_one_musicmaker_two],
459         [(9, 4), (10, 4), soprano_one_musicmaker_one],
460         [(10, 4), (12, 4), soprano_one_musicmaker_one],
461         [(16, 4), (17, 4), soprano_one_musicmaker_two],
462         [(17, 4), (18, 4), soprano_one_musicmaker_two],
463         [(18, 4), (19, 4), soprano_one_musicmaker_two],
464         [(21, 4), (23, 4), soprano_one_musicmaker_one],
465         [(23, 4), (24, 4), soprano_one_musicmaker_one],
466         [(24, 4), (25, 4), soprano_one_musicmaker_one],
467         [(25, 4), (27, 4), soprano_one_musicmaker_one],
468         [(29, 4), (30, 4), soprano_one_musicmaker_two],
469         [(30, 4), (31, 4), soprano_one_musicmaker_two],
470         [(31, 4), (32, 4), soprano_one_musicmaker_two],
471         [(36, 4), (37, 4), soprano_one_musicmaker_one],
472         [(37, 4), (38, 4), soprano_one_musicmaker_one],
473         [(38, 4), (39, 4), soprano_one_musicmaker_one],
474         [(41, 4), (42, 4), soprano_one_musicmaker_two],
475         [(42, 4), (43, 4), soprano_one_musicmaker_two],
476         [(43, 4), (87, 8), soprano_one_musicmaker_two],
477     ]
478 ])
479
480 voice_3_timespan_list = abjad.TimespanList([
481     abjad.AnnotatedTimespan(
482         start_offset=start_offset,
483         stop_offset=stop_offset,
484         annotation=MusicSpecifier(
485             music_maker=music_maker,
486             voice_name='Voice 3',
487         ),
488     )
489     for start_offset, stop_offset, music_maker in [
490         [(2, 4), (4, 4), soprano_two_musicmaker_one],
491         [(4, 4), (5, 4), soprano_two_musicmaker_one],
492         [(9, 4), (10, 4), soprano_two_musicmaker_two],
493         [(10, 4), (11, 4), soprano_two_musicmaker_two],

```

```

494     [(11, 4), (12, 4), soprano_two_musicmaker_two],
495     [(14, 4), (15, 4), soprano_two_musicmaker_two],
496     [(15, 4), (17, 4), soprano_two_musicmaker_two],
497     [(21, 4), (22, 4), soprano_two_musicmaker_one],
498     [(22, 4), (24, 4), soprano_two_musicmaker_one],
499     [(24, 4), (25, 4), soprano_two_musicmaker_two],
500     [(25, 4), (26, 4), soprano_two_musicmaker_two],
501     [(26, 4), (27, 4), soprano_two_musicmaker_two],
502     [(31, 4), (33, 4), soprano_two_musicmaker_one],
503     [(33, 4), (34, 4), soprano_two_musicmaker_one],
504     [(36, 4), (37, 4), soprano_two_musicmaker_one],
505     [(37, 4), (39, 4), soprano_two_musicmaker_one],
506     [(43, 4), (44, 4), soprano_two_musicmaker_two],
507     [(44, 4), (45, 4), soprano_two_musicmaker_two],
508     [(45, 4), (91, 8), soprano_two_musicmaker_two],
509   ]
510 ])
511
512 voice_4_timespan_list = abjad.TimespanList([
513     abjad.AnnotatedTimespan(
514         start_offset=start_offset,
515         stop_offset=stop_offset,
516         annotation=MusicSpecifier(
517             music_maker=music_maker,
518             voice_name='Voice 4',
519         ),
520     )
521     for start_offset, stop_offset, music_maker in [
522         [(1, 8), (1, 4), soprano_three_musicmaker_two],
523         [(1, 4), (2, 4), soprano_three_musicmaker_two],
524         [(2, 4), (3, 4), soprano_three_musicmaker_two],
525         [(7, 4), (8, 4), soprano_three_musicmaker_two],
526         [(8, 4), (9, 4), soprano_three_musicmaker_two],
527         [(9, 4), (10, 4), soprano_three_musicmaker_two],
528         [(25, 8), (13, 4), soprano_three_musicmaker_one],
529         [(13, 4), (14, 4), soprano_three_musicmaker_one],
530         [(14, 4), (15, 4), soprano_three_musicmaker_one],
531         [(19, 4), (20, 4), soprano_three_musicmaker_two],
532         [(20, 4), (21, 4), soprano_three_musicmaker_two],
533         [(21, 4), (22, 4), soprano_three_musicmaker_two],
534         [(26, 4), (27, 4), soprano_three_musicmaker_one],
535         [(27, 4), (28, 4), soprano_three_musicmaker_one],
536         [(28, 4), (29, 4), soprano_three_musicmaker_one],
537         [(33, 4), (34, 4), soprano_three_musicmaker_one],
538         [(34, 4), (35, 4), soprano_three_musicmaker_one],
539         [(35, 4), (36, 4), soprano_three_musicmaker_one],
540         [(38, 4), (39, 4), soprano_three_musicmaker_two],
541         [(39, 4), (40, 4), soprano_three_musicmaker_two],
542         [(40, 4), (41, 4), soprano_three_musicmaker_two],
543         [(45, 4), (46, 4), soprano_three_musicmaker_one],
544         [(46, 4), (47, 4), soprano_three_musicmaker_one],
545         [(47, 4), (95, 8), soprano_three_musicmaker_one],
546     ]
547 ])
548
549 voice_5_timespan_list = abjad.TimespanList([
550     abjad.AnnotatedTimespan(
551         start_offset=start_offset,
552         stop_offset=stop_offset,

```

```

553     annotation=MusicSpecifier(
554         music_maker=music_maker,
555         voice_name='Voice 5',
556     ),
557 )
558 for start_offset, stop_offset, music_maker in [
559     [(0, 4), (1, 4), alto_one_musicmaker_one],
560     [(1, 4), (2, 4), alto_one_musicmaker_one],
561     [(2, 4), (3, 4), alto_one_musicmaker_one],
562     [(5, 4), (6, 4), alto_one_musicmaker_one],
563     [(6, 4), (7, 4), alto_one_musicmaker_one],
564     [(7, 4), (8, 4), alto_one_musicmaker_one],
565     [(12, 4), (13, 4), alto_one_musicmaker_two],
566     [(13, 4), (14, 4), alto_one_musicmaker_two],
567     [(14, 4), (15, 4), alto_one_musicmaker_two],
568     [(17, 4), (18, 4), alto_one_musicmaker_one],
569     [(18, 4), (19, 4), alto_one_musicmaker_one],
570     [(19, 4), (20, 4), alto_one_musicmaker_one],
571     [(28, 4), (29, 4), alto_one_musicmaker_two],
572     [(29, 4), (30, 4), alto_one_musicmaker_two],
573     [(30, 4), (31, 4), alto_one_musicmaker_two],
574     [(33, 4), (34, 4), alto_one_musicmaker_two],
575     [(34, 4), (35, 4), alto_one_musicmaker_two],
576     [(35, 4), (36, 4), alto_one_musicmaker_two],
577     [(40, 4), (41, 4), alto_one_musicmaker_one],
578     [(41, 4), (42, 4), alto_one_musicmaker_one],
579     [(42, 4), (43, 4), alto_one_musicmaker_one],
580     [(45, 4), (46, 4), alto_one_musicmaker_two],
581     [(46, 4), (47, 4), alto_one_musicmaker_two],
582     [(47, 4), (95, 8), alto_one_musicmaker_two],
583 ]
584 ])
585
586 voice_6_timestspan_list = abjad.TimespanList([
587     abjad.AnnotatedTimespan(
588         start_offset=start_offset,
589         stop_offset=stop_offset,
590         annotation=MusicSpecifier(
591             music_maker=music_maker,
592             voice_name='Voice 6',
593         ),
594     )
595     for start_offset, stop_offset, music_maker in [
596         [(4, 4), (5, 4), alto_two_musicmaker_two],
597         [(5, 4), (6, 4), alto_two_musicmaker_two],
598         [(6, 4), (7, 4), alto_two_musicmaker_two],
599         [(9, 4), (10, 4), alto_two_musicmaker_one],
600         [(10, 4), (11, 4), alto_two_musicmaker_one],
601         [(11, 4), (12, 4), alto_two_musicmaker_one],
602         [(16, 4), (17, 4), alto_two_musicmaker_two],
603         [(17, 4), (18, 4), alto_two_musicmaker_two],
604         [(18, 4), (19, 4), alto_two_musicmaker_two],
605         [(21, 4), (22, 4), alto_two_musicmaker_one],
606         [(22, 4), (23, 4), alto_two_musicmaker_one],
607         [(23, 4), (24, 4), alto_two_musicmaker_one],
608         [(24, 4), (25, 4), alto_two_musicmaker_one],
609         [(25, 4), (26, 4), alto_two_musicmaker_one],
610         [(26, 4), (27, 4), alto_two_musicmaker_one],
611         [(29, 4), (30, 4), alto_two_musicmaker_two],

```

```

612     [(30, 4), (31, 4), alto_two_musicmaker_two],
613     [(31, 4), (32, 4), alto_two_musicmaker_two],
614     [(36, 4), (37, 4), alto_two_musicmaker_one],
615     [(37, 4), (38, 4), alto_two_musicmaker_one],
616     [(38, 4), (39, 4), alto_two_musicmaker_one],
617     [(41, 4), (42, 4), alto_two_musicmaker_two],
618     [(42, 4), (43, 4), alto_two_musicmaker_two],
619     [(43, 4), (87, 8), alto_two_musicmaker_two],
620 ]
621 ])
622
623 voice_7_timespan_list = abjad.TimespanList([
624     abjad.AnnotatedTimespan(
625         start_offset=start_offset,
626         stop_offset=stop_offset,
627         annotation=MusicSpecifier(
628             music_maker=music_maker,
629             voice_name='Voice 7',
630         ),
631     )
632     for start_offset, stop_offset, music_maker in [
633         [(2, 4), (3, 4), alto_three_musicmaker_one],
634         [(3, 4), (4, 4), alto_three_musicmaker_one],
635         [(4, 4), (5, 4), alto_three_musicmaker_one],
636         [(9, 4), (10, 4), alto_three_musicmaker_two],
637         [(10, 4), (11, 4), alto_three_musicmaker_two],
638         [(11, 4), (12, 4), alto_three_musicmaker_two],
639         [(14, 4), (15, 4), alto_three_musicmaker_two],
640         [(15, 4), (16, 4), alto_three_musicmaker_two],
641         [(16, 4), (17, 4), alto_three_musicmaker_two],
642         [(21, 4), (22, 4), alto_three_musicmaker_one],
643         [(22, 4), (23, 4), alto_three_musicmaker_one],
644         [(23, 4), (24, 4), alto_three_musicmaker_one],
645         [(24, 4), (25, 4), alto_three_musicmaker_two],
646         [(25, 4), (26, 4), alto_three_musicmaker_two],
647         [(26, 4), (27, 4), alto_three_musicmaker_two],
648         [(31, 4), (32, 4), alto_three_musicmaker_one],
649         [(32, 4), (33, 4), alto_three_musicmaker_one],
650         [(33, 4), (34, 4), alto_three_musicmaker_one],
651         [(36, 4), (37, 4), alto_three_musicmaker_one],
652         [(37, 4), (38, 4), alto_three_musicmaker_one],
653         [(38, 4), (39, 4), alto_three_musicmaker_one],
654         [(43, 4), (44, 4), alto_three_musicmaker_two],
655         [(44, 4), (45, 4), alto_three_musicmaker_two],
656         [(45, 4), (91, 8), alto_three_musicmaker_two],
657     ]
658 ])
659
660 voice_8_timespan_list = abjad.TimespanList([
661     abjad.AnnotatedTimespan(
662         start_offset=start_offset,
663         stop_offset=stop_offset,
664         annotation=MusicSpecifier(
665             music_maker=music_maker,
666             voice_name='Voice 8',
667         ),
668     )
669     for start_offset, stop_offset, music_maker in [
670         [(1, 8), (3, 4), alto_four_musicmaker_two],

```

```

671     [(7, 4), (9, 4), alto_four_musicmaker_two],
672     [(9, 4), (10, 4), alto_four_musicmaker_two],
673     [(25, 8), (13, 4), alto_four_musicmaker_one],
674     [(13, 4), (14, 4), alto_four_musicmaker_one],
675     [(14, 4), (15, 4), alto_four_musicmaker_one],
676     [(19, 4), (20, 4), alto_four_musicmaker_two],
677     [(20, 4), (21, 4), alto_four_musicmaker_two],
678     [(21, 4), (22, 4), alto_four_musicmaker_two],
679     [(26, 4), (27, 4), alto_four_musicmaker_one],
680     [(27, 4), (28, 4), alto_four_musicmaker_one],
681     [(28, 4), (29, 4), alto_four_musicmaker_one],
682     [(33, 4), (34, 4), alto_four_musicmaker_one],
683     [(34, 4), (35, 4), alto_four_musicmaker_one],
684     [(35, 4), (36, 4), alto_four_musicmaker_one],
685     [(38, 4), (39, 4), alto_four_musicmaker_two],
686     [(39, 4), (40, 4), alto_four_musicmaker_two],
687     [(40, 4), (41, 4), alto_four_musicmaker_two],
688     [(45, 4), (46, 4), alto_four_musicmaker_one],
689     [(46, 4), (47, 4), alto_four_musicmaker_one],
690     [(47, 4), (95, 8), alto_four_musicmaker_one],
691   ]
692 ])
693
694 voice_9_timespan_list = abjad.TimespanList([
695     abjad.AnnotatedTimespan(
696         start_offset=start_offset,
697         stop_offset=stop_offset,
698         annotation=MusicSpecifier(
699             music_maker=music_maker,
700             voice_name='Voice 9',
701         ),
702     ),
703     for start_offset, stop_offset, music_maker in [
704         [(0, 4), (1, 4), alto_five_musicmaker_one],
705         [(1, 4), (3, 4), alto_five_musicmaker_one],
706         [(5, 4), (7, 4), alto_five_musicmaker_one],
707         [(7, 4), (8, 4), alto_five_musicmaker_one],
708         [(12, 4), (13, 4), alto_five_musicmaker_two],
709         [(13, 4), (15, 4), alto_five_musicmaker_two],
710         [(17, 4), (19, 4), alto_five_musicmaker_one],
711         [(19, 4), (20, 4), alto_five_musicmaker_one],
712         [(28, 4), (29, 4), alto_five_musicmaker_two],
713         [(29, 4), (31, 4), alto_five_musicmaker_two],
714         [(33, 4), (35, 4), alto_five_musicmaker_two],
715         [(35, 4), (36, 4), alto_five_musicmaker_two],
716         [(40, 4), (41, 4), alto_five_musicmaker_one],
717         [(41, 4), (42, 4), alto_five_musicmaker_one],
718         [(42, 4), (43, 4), alto_five_musicmaker_one],
719         [(45, 4), (46, 4), alto_five_musicmaker_two],
720         [(46, 4), (47, 4), alto_five_musicmaker_two],
721         [(47, 4), (95, 8), alto_five_musicmaker_two],
722     ]
723   ])
724
725 voice_10_timespan_list = abjad.TimespanList([
726     abjad.AnnotatedTimespan(
727         start_offset=start_offset,
728         stop_offset=stop_offset,
729         annotation=MusicSpecifier(

```

```

730     music_maker=music_maker,
731     voice_name='Voice 10',
732   ),
733 )
734 for start_offset, stop_offset, music_maker in [
735   [(4, 4), (5, 4), alto_six_musicmaker_two],
736   [(5, 4), (7, 4), alto_six_musicmaker_two],
737   [(9, 4), (11, 4), alto_six_musicmaker_one],
738   [(11, 4), (12, 4), alto_six_musicmaker_one],
739   [(16, 4), (17, 4), alto_six_musicmaker_two],
740   [(17, 4), (19, 4), alto_six_musicmaker_two],
741   [(21, 4), (23, 4), alto_six_musicmaker_one],
742   [(23, 4), (24, 4), alto_six_musicmaker_one],
743   [(24, 4), (26, 4), alto_six_musicmaker_one],
744   [(26, 4), (27, 4), alto_six_musicmaker_one],
745   [(29, 4), (31, 4), alto_six_musicmaker_two],
746   [(31, 4), (32, 4), alto_six_musicmaker_two],
747   [(36, 4), (38, 4), alto_six_musicmaker_one],
748   [(38, 4), (39, 4), alto_six_musicmaker_one],
749   [(41, 4), (42, 4), alto_six_musicmaker_two],
750   [(42, 4), (43, 4), alto_six_musicmaker_two],
751   [(43, 4), (87, 8), alto_six_musicmaker_two],
752 ]
753 ])
754
755 voice_11_timespan_list = abjad.TimespanList([
756   abjad.AnnotatedTimespan(
757     start_offset=start_offset,
758     stop_offset=stop_offset,
759     annotation=MusicSpecifier(
760       music_maker=music_maker,
761       voice_name='Voice 11',
762     ),
763   ),
764   for start_offset, stop_offset, music_maker in [
765     [(2, 4), (4, 4), tenor_one_musicmaker_one],
766     [(4, 4), (5, 4), tenor_one_musicmaker_one],
767     [(9, 4), (10, 4), tenor_one_musicmaker_two],
768     [(10, 4), (12, 4), tenor_one_musicmaker_two],
769     [(14, 4), (16, 4), tenor_one_musicmaker_two],
770     [(16, 4), (17, 4), tenor_one_musicmaker_two],
771     [(21, 4), (22, 4), tenor_one_musicmaker_one],
772     [(22, 4), (24, 4), tenor_one_musicmaker_one],
773     [(24, 4), (25, 4), tenor_one_musicmaker_two],
774     [(25, 4), (26, 4), tenor_one_musicmaker_two],
775     [(26, 4), (27, 4), tenor_one_musicmaker_two],
776     [(31, 4), (33, 4), tenor_one_musicmaker_one],
777     [(33, 4), (34, 4), tenor_one_musicmaker_one],
778     [(36, 4), (37, 4), tenor_one_musicmaker_one],
779     [(37, 4), (39, 4), tenor_one_musicmaker_one],
780     [(43, 4), (45, 4), tenor_one_musicmaker_two],
781     [(45, 4), (91, 8), tenor_one_musicmaker_two],
782   ]
783 ])
784
785 voice_12_timespan_list = abjad.TimespanList([
786   abjad.AnnotatedTimespan(
787     start_offset=start_offset,
788     stop_offset=stop_offset,

```

```

789     annotation=MusicSpecifier(
790         music_maker=music_maker,
791         voice_name='Voice 12',
792     ),
793 )
794 for start_offset, stop_offset, music_maker in [
795     [(1, 8), (2, 4), tenor_two_musicmaker_two],
796     [(2, 4), (3, 4), tenor_two_musicmaker_two],
797     [(7, 4), (8, 4), tenor_two_musicmaker_two],
798     [(8, 4), (9, 4), tenor_two_musicmaker_two],
799     [(9, 4), (10, 4), tenor_two_musicmaker_two],
800     [(25, 8), (14, 4), tenor_two_musicmaker_one],
801     [(14, 4), (15, 4), tenor_two_musicmaker_one],
802     [(19, 4), (20, 4), tenor_two_musicmaker_two],
803     [(20, 4), (21, 4), tenor_two_musicmaker_two],
804     [(21, 4), (22, 4), tenor_two_musicmaker_two],
805     [(26, 4), (27, 4), tenor_two_musicmaker_one],
806     [(27, 4), (28, 4), tenor_two_musicmaker_one],
807     [(28, 4), (29, 4), tenor_two_musicmaker_one],
808     [(33, 4), (34, 4), tenor_two_musicmaker_one],
809     [(34, 4), (36, 4), tenor_two_musicmaker_one],
810     [(38, 4), (39, 4), tenor_two_musicmaker_two],
811     [(39, 4), (40, 4), tenor_two_musicmaker_two],
812     [(40, 4), (41, 4), tenor_two_musicmaker_two],
813     [(45, 4), (47, 4), tenor_two_musicmaker_one],
814     [(47, 4), (95, 8), tenor_two_musicmaker_one],
815 ]
816 ])
817
818 voice_13_timespan_list = abjad.TimespanList([
819     abjad.AnnotatedTimespan(
820         start_offset=start_offset,
821         stop_offset=stop_offset,
822         annotation=MusicSpecifier(
823             music_maker=music_maker,
824             voice_name='Voice 13',
825         ),
826     ),
827     for start_offset, stop_offset, music_maker in [
828         [(0, 4), (1, 4), tenor_three_musicmaker_one],
829         [(1, 4), (3, 4), tenor_three_musicmaker_one],
830         [(5, 4), (7, 4), tenor_three_musicmaker_one],
831         [(7, 4), (8, 4), tenor_three_musicmaker_one],
832         [(12, 4), (13, 4), tenor_three_musicmaker_two],
833         [(13, 4), (15, 4), tenor_three_musicmaker_two],
834         [(17, 4), (19, 4), tenor_three_musicmaker_one],
835         [(19, 4), (20, 4), tenor_three_musicmaker_one],
836         [(28, 4), (29, 4), tenor_three_musicmaker_two],
837         [(29, 4), (31, 4), tenor_three_musicmaker_two],
838         [(33, 4), (35, 4), tenor_three_musicmaker_two],
839         [(35, 4), (36, 4), tenor_three_musicmaker_two],
840         [(40, 4), (41, 4), tenor_three_musicmaker_one],
841         [(41, 4), (43, 4), tenor_three_musicmaker_one],
842         [(45, 4), (46, 4), tenor_three_musicmaker_two],
843         [(46, 4), (47, 4), tenor_three_musicmaker_two],
844         [(47, 4), (95, 8), tenor_three_musicmaker_two],
845     ]
846 ])
847

```

```

848 voice_14_timespan_list = abjad.TimespanList([
849     abjad.AnnotatedTimespan(
850         start_offset=start_offset,
851         stop_offset=stop_offset,
852         annotation=MusicSpecifier(
853             music_maker=music_maker,
854             voice_name='Voice 14',
855         ),
856     )
857     for start_offset, stop_offset, music_maker in [
858         [(4, 4), (5, 4), tenor_four_musicmaker_two],
859         [(5, 4), (6, 4), tenor_four_musicmaker_two],
860         [(6, 4), (7, 4), tenor_four_musicmaker_two],
861         [(9, 4), (10, 4), tenor_four_musicmaker_one],
862         [(10, 4), (11, 4), tenor_four_musicmaker_one],
863         [(11, 4), (12, 4), tenor_four_musicmaker_one],
864         [(16, 4), (17, 4), tenor_four_musicmaker_two],
865         [(17, 4), (18, 4), tenor_four_musicmaker_two],
866         [(18, 4), (19, 4), tenor_four_musicmaker_two],
867         [(21, 4), (22, 4), tenor_four_musicmaker_one],
868         [(22, 4), (24, 4), tenor_four_musicmaker_one],
869         [(24, 4), (26, 4), tenor_four_musicmaker_one],
870         [(26, 4), (27, 4), tenor_four_musicmaker_one],
871         [(29, 4), (30, 4), tenor_four_musicmaker_two],
872         [(30, 4), (31, 4), tenor_four_musicmaker_two],
873         [(31, 4), (32, 4), tenor_four_musicmaker_two],
874         [(36, 4), (38, 4), tenor_four_musicmaker_one],
875         [(38, 4), (39, 4), tenor_four_musicmaker_one],
876         [(41, 4), (42, 4), tenor_four_musicmaker_two],
877         [(42, 4), (43, 4), tenor_four_musicmaker_two],
878         [(43, 4), (87, 8), tenor_four_musicmaker_two],
879     ]
880 ])
881
882 voice_15_timespan_list = abjad.TimespanList([
883     abjad.AnnotatedTimespan(
884         start_offset=start_offset,
885         stop_offset=stop_offset,
886         annotation=MusicSpecifier(
887             music_maker=music_maker,
888             voice_name='Voice 15',
889         ),
890     )
891     for start_offset, stop_offset, music_maker in [
892         [(2, 4), (3, 4), tenor_five_musicmaker_one],
893         [(3, 4), (5, 4), tenor_five_musicmaker_one],
894         [(9, 4), (11, 4), tenor_five_musicmaker_two],
895         [(11, 4), (12, 4), tenor_five_musicmaker_two],
896         [(14, 4), (15, 4), tenor_five_musicmaker_two],
897         [(15, 4), (17, 4), tenor_five_musicmaker_two],
898         [(21, 4), (23, 4), tenor_five_musicmaker_one],
899         [(23, 4), (24, 4), tenor_five_musicmaker_one],
900         [(24, 4), (26, 4), tenor_five_musicmaker_two],
901         [(26, 4), (27, 4), tenor_five_musicmaker_two],
902         [(31, 4), (32, 4), tenor_five_musicmaker_one],
903         [(32, 4), (34, 4), tenor_five_musicmaker_one],
904         [(36, 4), (38, 4), tenor_five_musicmaker_one],
905         [(38, 4), (39, 4), tenor_five_musicmaker_one],
906         [(43, 4), (44, 4), tenor_five_musicmaker_two],

```

```

907     [(44, 4), (45, 4), tenor_five_musicmaker_two],
908     [(45, 4), (91, 8), tenor_five_musicmaker_two],
909   ]
910 ])
911
912 voice_16_timespan_list = abjad.TimespanList([
913     abjad.AnnotatedTimespan(
914         start_offset=start_offset,
915         stop_offset=stop_offset,
916         annotation=MusicSpecifier(
917             music_maker=music_maker,
918             voice_name='Voice 16',
919         ),
920     ),
921     for start_offset, stop_offset, music_maker in [
922         [(1, 8), (2, 4), baritone_one_musicmaker_two],
923         [(2, 4), (3, 4), baritone_one_musicmaker_two],
924         [(7, 4), (8, 4), baritone_one_musicmaker_two],
925         [(8, 4), (9, 4), baritone_one_musicmaker_two],
926         [(9, 4), (10, 4), baritone_one_musicmaker_two],
927         [(25, 8), (13, 4), baritone_one_musicmaker_one],
928         [(13, 4), (15, 4), baritone_one_musicmaker_one],
929         [(19, 4), (21, 4), baritone_one_musicmaker_two],
930         [(21, 4), (22, 4), baritone_one_musicmaker_two],
931         [(26, 4), (27, 4), baritone_one_musicmaker_one],
932         [(27, 4), (29, 4), baritone_one_musicmaker_one],
933         [(33, 4), (35, 4), baritone_one_musicmaker_one],
934         [(35, 4), (36, 4), baritone_one_musicmaker_one],
935         [(38, 4), (39, 4), baritone_one_musicmaker_two],
936         [(39, 4), (41, 4), baritone_one_musicmaker_two],
937         [(45, 4), (46, 4), baritone_one_musicmaker_one],
938         [(46, 4), (47, 4), baritone_one_musicmaker_one],
939         [(47, 4), (95, 8), baritone_one_musicmaker_one],
940     ],
941   ],
942 ]
943 voice_17_timespan_list = abjad.TimespanList([
944     abjad.AnnotatedTimespan(
945         start_offset=start_offset,
946         stop_offset=stop_offset,
947         annotation=MusicSpecifier(
948             music_maker=music_maker,
949             voice_name='Voice 17',
950         ),
951     ),
952     for start_offset, stop_offset, music_maker in [
953         [(0, 4), (1, 4), baritone_two_musicmaker_one],
954         [(1, 4), (2, 4), baritone_two_musicmaker_one],
955         [(2, 4), (3, 4), baritone_two_musicmaker_one],
956         [(5, 4), (6, 4), baritone_two_musicmaker_one],
957         [(6, 4), (8, 4), baritone_two_musicmaker_one],
958         [(12, 4), (14, 4), baritone_two_musicmaker_two],
959         [(14, 4), (15, 4), baritone_two_musicmaker_two],
960         [(17, 4), (18, 4), baritone_two_musicmaker_one],
961         [(18, 4), (20, 4), baritone_two_musicmaker_one],
962         [(28, 4), (30, 4), baritone_two_musicmaker_two],
963         [(30, 4), (31, 4), baritone_two_musicmaker_two],
964         [(33, 4), (34, 4), baritone_two_musicmaker_two],
965         [(34, 4), (36, 4), baritone_two_musicmaker_two],

```

```

966     [(40, 4), (42, 4), baritone_two_musicmaker_one],
967     [(42, 4), (43, 4), baritone_two_musicmaker_one],
968     [(45, 4), (47, 4), baritone_two_musicmaker_two],
969     [(47, 4), (95, 8), baritone_two_musicmaker_two],
970   ]
971 ])
972
973 voice_18_timespan_list = abjad.TimespanList([
974     abjad.AnnotatedTimespan(
975         start_offset=start_offset,
976         stop_offset=stop_offset,
977         annotation=MusicSpecifier(
978             music_maker=music_maker,
979             voice_name='Voice 18',
980         ),
981     )
982     for start_offset, stop_offset, music_maker in [
983         [(4, 4), (5, 4), baritone_three_musicmaker_two],
984         [(5, 4), (7, 4), baritone_three_musicmaker_two],
985         [(9, 4), (11, 4), baritone_three_musicmaker_one],
986         [(11, 4), (12, 4), baritone_three_musicmaker_one],
987         [(16, 4), (17, 4), baritone_three_musicmaker_two],
988         [(17, 4), (19, 4), baritone_three_musicmaker_two],
989         [(21, 4), (23, 4), baritone_three_musicmaker_one],
990         [(23, 4), (24, 4), baritone_three_musicmaker_one],
991         [(24, 4), (25, 4), baritone_three_musicmaker_one],
992         [(25, 4), (27, 4), baritone_three_musicmaker_one],
993         [(29, 4), (31, 4), baritone_three_musicmaker_two],
994         [(31, 4), (32, 4), baritone_three_musicmaker_two],
995         [(36, 4), (37, 4), baritone_three_musicmaker_one],
996         [(37, 4), (39, 4), baritone_three_musicmaker_one],
997         [(41, 4), (43, 4), baritone_three_musicmaker_two],
998         [(43, 4), (87, 8), baritone_three_musicmaker_two],
999     ]
1000   ])
1001
1002 voice_19_timespan_list = abjad.TimespanList([
1003     abjad.AnnotatedTimespan(
1004         start_offset=start_offset,
1005         stop_offset=stop_offset,
1006         annotation=MusicSpecifier(
1007             music_maker=music_maker,
1008             voice_name='Voice 19',
1009         ),
1010     )
1011     for start_offset, stop_offset, music_maker in [
1012         [(2, 4), (4, 4), bass_one_musicmaker_one],
1013         [(4, 4), (5, 4), bass_one_musicmaker_one],
1014         [(9, 4), (11, 4), bass_one_musicmaker_two],
1015         [(11, 4), (12, 4), bass_one_musicmaker_two],
1016         [(14, 4), (15, 4), bass_one_musicmaker_two],
1017         [(15, 4), (17, 4), bass_one_musicmaker_two],
1018         [(21, 4), (22, 4), bass_one_musicmaker_one],
1019         [(22, 4), (24, 4), bass_one_musicmaker_one],
1020         [(24, 4), (25, 4), bass_one_musicmaker_two],
1021         [(25, 4), (26, 4), bass_one_musicmaker_two],
1022         [(26, 4), (27, 4), bass_one_musicmaker_two],
1023         [(31, 4), (33, 4), bass_one_musicmaker_one],
1024         [(33, 4), (34, 4), bass_one_musicmaker_one],

```

```

1025     [(36, 4), (37, 4), bass_one_musicmaker_one],
1026     [(37, 4), (39, 4), bass_one_musicmaker_one],
1027     [(43, 4), (44, 4), bass_one_musicmaker_two],
1028     [(44, 4), (91, 8), bass_one_musicmaker_two],
1029 ]
1030 ])
1031
1032 voice_20_timespan_list = abjad.TimespanList([
1033     abjad.AnnotatedTimespan(
1034         start_offset=start_offset,
1035         stop_offset=stop_offset,
1036         annotation=MusicSpecifier(
1037             music_maker=music_maker,
1038             voice_name='Voice 20',
1039         ),
1040     )
1041     for start_offset, stop_offset, music_maker in [
1042         [(1, 8), (1, 4), bass_two_musicmaker_two],
1043         [(1, 4), (2, 4), bass_two_musicmaker_two],
1044         [(2, 4), (3, 4), bass_two_musicmaker_two],
1045         [(7, 4), (8, 4), bass_two_musicmaker_two],
1046         [(8, 4), (9, 4), bass_two_musicmaker_two],
1047         [(9, 4), (10, 4), bass_two_musicmaker_two],
1048         [(25, 8), (13, 4), bass_two_musicmaker_one],
1049         [(13, 4), (14, 4), bass_two_musicmaker_one],
1050         [(14, 4), (15, 4), bass_two_musicmaker_one],
1051         [(19, 4), (20, 4), bass_two_musicmaker_two],
1052         [(20, 4), (21, 4), bass_two_musicmaker_two],
1053         [(21, 4), (22, 4), bass_two_musicmaker_two],
1054         [(26, 4), (27, 4), bass_two_musicmaker_one],
1055         [(27, 4), (28, 4), bass_two_musicmaker_one],
1056         [(28, 4), (29, 4), bass_two_musicmaker_one],
1057         [(33, 4), (34, 4), bass_two_musicmaker_one],
1058         [(34, 4), (35, 4), bass_two_musicmaker_one],
1059         [(35, 4), (36, 4), bass_two_musicmaker_one],
1060         [(38, 4), (39, 4), bass_two_musicmaker_two],
1061         [(39, 4), (40, 4), bass_two_musicmaker_two],
1062         [(40, 4), (41, 4), bass_two_musicmaker_two],
1063         [(45, 4), (46, 4), bass_two_musicmaker_one],
1064         [(46, 4), (47, 4), bass_two_musicmaker_one],
1065         [(47, 4), (95, 8), bass_two_musicmaker_one],
1066     ]
1067 ])
1068
1069 voice_21_timespan_list = abjad.TimespanList([
1070     abjad.AnnotatedTimespan(
1071         start_offset=start_offset,
1072         stop_offset=stop_offset,
1073         annotation=MusicSpecifier(
1074             music_maker=music_maker,
1075             voice_name='Voice 21',
1076         ),
1077     )
1078     for start_offset, stop_offset, music_maker in [
1079         [(0, 4), (2, 4), contrabass_musicmaker_one],
1080         [(2, 4), (3, 4), contrabass_musicmaker_one],
1081         [(5, 4), (7, 4), contrabass_musicmaker_one],
1082         [(7, 4), (8, 4), contrabass_musicmaker_one],
1083         [(12, 4), (14, 4), contrabass_musicmaker_two],

```

```

1084 [(14, 4), (15, 4), contrabass_musicmaker_two],
1085 [(17, 4), (18, 4), contrabass_musicmaker_one],
1086 [(18, 4), (20, 4), contrabass_musicmaker_one],
1087 [(28, 4), (31, 4), contrabass_musicmaker_two],
1088 [(33, 4), (35, 4), contrabass_musicmaker_two],
1089 [(35, 4), (36, 4), contrabass_musicmaker_two],
1090 [(40, 4), (42, 4), contrabass_musicmaker_one],
1091 [(42, 4), (43, 4), contrabass_musicmaker_one],
1092 [(45, 4), (46, 4), contrabass_musicmaker_two],
1093 [(46, 4), (47, 4), contrabass_musicmaker_two],
1094 [(47, 4), (95, 8), contrabass_musicmaker_two],
1095 ]
1096 ])
1097
1098 all_timestspan_lists = {
1099     'Voice 1': voice_1_timestspan_list,
1100     'Voice 2': voice_2_timestspan_list,
1101     'Voice 3': voice_3_timestspan_list,
1102     'Voice 4': voice_4_timestspan_list,
1103     'Voice 5': voice_5_timestspan_list,
1104     'Voice 6': voice_6_timestspan_list,
1105     'Voice 7': voice_7_timestspan_list,
1106     'Voice 8': voice_8_timestspan_list,
1107     'Voice 9': voice_9_timestspan_list,
1108     'Voice 10': voice_10_timestspan_list,
1109     'Voice 11': voice_11_timestspan_list,
1110     'Voice 12': voice_12_timestspan_list,
1111     'Voice 13': voice_13_timestspan_list,
1112     'Voice 14': voice_14_timestspan_list,
1113     'Voice 15': voice_15_timestspan_list,
1114     'Voice 16': voice_16_timestspan_list,
1115     'Voice 17': voice_17_timestspan_list,
1116     'Voice 18': voice_18_timestspan_list,
1117     'Voice 19': voice_19_timestspan_list,
1118     'Voice 20': voice_20_timestspan_list,
1119     'Voice 21': voice_21_timestspan_list,
1120 }
1121
1122 global_timestspan = abjad.Timespan(
1123     start_offset=0,
1124     stop_offset=max(_.stop_offset for _ in all_timestspan_lists.values())
1125 )
1126
1127 for voice_name, timestspan_list in all_timestspan_lists.items():
1128     silences = abjad.TimespanList([global_timestspan])
1129     silences.extend(timestspan_list)
1130     silences.sort()
1131     silences.compute_logical_xor()
1132     for silence_timestspan in silences:
1133         timestspan_list.append(
1134             abjad.AnnotatedTimespan(
1135                 start_offset=silence_timestspan.start_offset,
1136                 stop_offset=silence_timestspan.stop_offset,
1137                 annotation=MusicSpecifier(
1138                     music_maker=None,
1139                     voice_name=voice_name,
1140                 ),
1141             ),
1142         )

```

```

1143 timespan_list.sort()
1144
1145 for voice_name, timespan_list in all_timespan_lists.items():
1146     shards = timespan_list.split_at_offsets(bounds)
1147     split_timespan_list = abjad.TimespanList()
1148     for shard in shards:
1149         split_timespan_list.extend(shard)
1150     split_timespan_list.sort()
1151     all_timespan_lists[voice_name] = timespan_list
1152
1153 score = abjad.Score([
1154     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
1155     abjad.StaffGroup(
1156         [
1157             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
1158                         lilypond_type='Staff',),
1159             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
1160                         lilypond_type='Staff',),
1161             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
1162                         lilypond_type='Staff',),
1163             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
1164                         lilypond_type='Staff',),
1165             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
1166                         lilypond_type='Staff',),
1167             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
1168                         lilypond_type='Staff',),
1169             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
1170                         lilypond_type='Staff',),
1171             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
1172                         lilypond_type='Staff',),
1173             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
1174                         lilypond_type='Staff',),
1175             abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
1176                         lilypond_type='Staff',),
1177             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
1178                         lilypond_type='Staff',),
1179             abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
1180                         lilypond_type='Staff',),
1181             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
1182                         lilypond_type='Staff',),
1183             abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
1184                         lilypond_type='Staff',),
1185             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
1186                         lilypond_type='Staff',),
1187             abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
1188                         lilypond_type='Staff',),
1189             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
1190                         lilypond_type='Staff',),
1191             abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
1192                         lilypond_type='Staff',),
1193             abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
1194                         lilypond_type='Staff',),
1195             abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
1196                         lilypond_type='Staff',),
1197             abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
1198                         lilypond_type='Staff',),
1199         ],
1200         name='Staff Group',
1201     )

```

```

1181 ])
1182
1183 for time_signature in time_signatures:
1184     skip = abjad.Skip(1, multiplier=(time_signature))
1185     abjad.attach(time_signature, skip)
1186     score['Global Context'].append(skip)
1187
1188 print('Making containers ...')
1189
1190 def make_container(music_maker, durations):
1191     selections = music_maker(durations)
1192     container = abjad.Container([])
1193     container.extend(selections)
1194     return container
1195
1196 def key_function(timespan):
1197     return timespan.annotation.music_maker or silence_maker
1198
1199 for voice_name, timespan_list in all_timespan_lists.items():
1200     for music_maker, grouper in itertools.groupby(
1201         timespan_list,
1202         key=key_function,
1203     ):
1204         durations = [timespan.duration for timespan in grouper]
1205         container = make_container(music_maker, durations)
1206         voice = score[voice_name]
1207         voice.append(container)
1208
1209 print('Splitting and rewriting ...')
1210 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
1211     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
1212         time_signature = time_signatures[i]
1213         abjad.mutate(shard).rewrite_meter(time_signature)
1214
1215 print('Beaming runs ...')
1216 for voice in abjad.select(score).components(abjad.Voice):
1217     for run in abjad.select(voice).runs():
1218         specifier = abjadext.rmakers.BeamSpecifier(
1219             beam_each_division=False,
1220         )
1221         specifier(run)
1222         abjad.beam(voice[:, beam_lone_notes=False, beam_rests=False])
1223
1224 print('Stopping Hairpins ...')
1225 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1226     for rest in abjad.iterate(staff).components(abjad.Rest):
1227         previous_leaf = abjad.inspect(rest).leaf(-1)
1228         if isinstance(previous_leaf, abjad.Note):
1229             abjad.attach(abjad.StopHairpin(), rest)
1230         elif isinstance(previous_leaf, abjad.Chord):
1231             abjad.attach(abjad.StopHairpin(), rest)
1232         elif isinstance(previous_leaf, abjad.Rest):
1233             pass
1234
1235 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1236     first_leaf = abjad.select(staff).leaves()[0]
1237     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
1238     abjad.attach(stop, first_leaf)
1239

```

```

1240 scales = [
1241     soprano_1_notes,
1242     soprano_2_notes,
1243     soprano_3_notes,
1244     alto_1_notes,
1245     alto_2_notes,
1246     alto_3_notes,
1247     alto_4_notes,
1248     alto_5_notes,
1249     alto_6_notes,
1250     tenor_1_notes,
1251     tenor_2_notes,
1252     tenor_3_notes,
1253     tenor_4_notes,
1254     tenor_5_notes,
1255     baritone_1_notes,
1256     baritone_2_notes,
1257     baritone_3_notes,
1258     bass_1_notes,
1259     bass_2_notes,
1260     contrabass_notes,
1261 ]
1262 ]
1263
1264 staffs = [staff for staff in abjad.iterate(score['Staff Group']).components(abjad.
1265 Staff)]
1266
1267 for staff, scale in zip(staffs, scales):
1268     logicl_ties = [i for i in abjad.iterate(staff).logical_ties(pitched=True)]
1269     pitches = cyc(scale)
1270     for i, logicl_tie in enumerate(logicl_ties):
1271         if logicl_tie.is_pitched == True:
1272             pitch = next(pitches)
1273             for note in logicl_tie:
1274                 note.written_pitch = pitch
1275
1276 print('Adding attachments ...')
1277 bar_line = abjad.BarLine('||')
1278 metro = abjad.MetronomeMark((1, 4), 60)
1279 markup = abjad.Markup(r'\bold { H }')
1280 mark = abjad.RehearsalMark(markup=markup)
1281
1282 instruments = cyc([
1283     abjad.SopranoSaxophone(),
1284     abjad.SopranoSaxophone(),
1285     abjad.SopranoSaxophone(),
1286     abjad.SopranoSaxophone(),
1287     abjad.AltoSaxophone(),
1288     abjad.AltoSaxophone(),
1289     abjad.AltoSaxophone(),
1290     abjad.AltoSaxophone(),
1291     abjad.AltoSaxophone(),
1292     abjad.TenorSaxophone(),
1293     abjad.TenorSaxophone(),
1294     abjad.TenorSaxophone(),
1295     abjad.TenorSaxophone(),
1296     abjad.TenorSaxophone(),
1297     abjad.BaritoneSaxophone(),

```

```

1298 abjad.BaritoneSaxophone(),
1299 abjad.BaritoneSaxophone(),
1300 abjad.BassSaxophone(),
1301 abjad.BassSaxophone(),
1302 abjad.ContrabassSaxophone(),
1303 ])
1304
1305 abbreviations = cyc([
1306     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
1307     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
1308     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
1309     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
1310     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
1311     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
1312     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
1313     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
1314     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
1315     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
1316     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
1317     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
1318     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
1319     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
1320     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
1321     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
1322     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
1323     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
1324     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
1325     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
1326     abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
1327 ])
1328
1329 names = cyc([
1330     abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
1331     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
1332     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
1333     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
1334     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
1335     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
1336     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
1337     abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
1338     abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
1339     abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
1340     abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
1341     abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
1342     abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
1343     abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
1344     abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
1345     abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
1346     abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
1347     abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),
1348     abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
1349     abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
1350     abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
1351 ])
1352
1353 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1354     leaf1 = abjad.select(staff).leaves()[0]
1355     abjad.attach(next(instruments), leaf1)
1356     abjad.attach(next(abbreviations), leaf1)

```

```

1357 abjad.attach(next(names), leaf1)
1358
1359 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
1360     leaf1 = abjad.select(staff).leaves()[0]
1361     last_leaf = abjad.select(staff).leaves()[-1]
1362     abjad.attach(metro, leaf1)
1363     abjad.attach(bar_line, last_leaf)
1364
1365 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
1366     leaf1 = abjad.select(staff).leaves()[0]
1367     abjad.attach(mark, leaf1)
1368
1369 score_file = abjad.LilyPondFile.new(
1370     score,
1371     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1372     _stylesheets/abjad.ily'],
1373 )
1374
1375 abjad.SegmentMaker.comment_measure_numbers(score)
1376 ######
1377
1378 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_H'
1379 pdf_path = f'{directory}/Section_H.pdf'
1380 path = pathlib.Path('Section_H.pdf')
1381 if path.exists():
1382     print(f'Removing {pdf_path} ...')
1383     path.unlink()
1384 time_1 = time.time()
1385 print(f'Persisting {pdf_path} ...')
1386 result = abjad.persist(score_file).as_pdf(pdf_path)
1387 print(result[0])
1388 print(result[1])
1389 print(result[2])
1390 success = result[3]
1391 if success is False:
1392     print('LilyPond failed!')
1393 time_2 = time.time()
1394 total_time = time_2 - time_1
1395 print(f'Total time: {total_time} seconds')
1396 if path.exists():
1397     print(f'Opening {pdf_path} ...')
1398     os.system(f'open {pdf_path}')
1399 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_H/Section_H.
1400 ly').readlines()
1401 open('/Users/evansdsg2/Scores/guerrero/Build/Section_H.ly', 'w').writelines(
1402     score_lines[15:-1])
1403
1404 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
1405     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1406     signature_copy = abjad.mutate(signatures).copy()
1407     staff_copy = abjad.mutate(staff).copy()
1408     part = abjad.Score()
1409     part.insert(0, staff)
1410     part.insert(0, signature_copy)
1411     part_file = abjad.LilyPondFile.new(
1412         part,
1413         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1414         _stylesheets/abjad.ily'],
1415     )

```

```

1412 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.) sopranino'
1413 pdf_path = f'{directory}/Section_H.pdf'
1414 path = pathlib.Path('Section_H.pdf')
1415 if path.exists():
1416     print(f'Removing {pdf_path} ...')
1417     path.unlink()
1418 time_1 = time.time()
1419 print(f'Persisting {pdf_path} ...')
1420 result = abjad.persist(part_file).as_pdf(pdf_path)
1421 print(result[0])
1422 print(result[1])
1423 print(result[2])
1424 success = result[3]
1425 if success is False:
1426     print('LilyPond failed!')
1427 time_2 = time.time()
1428 total_time = time_2 - time_1
1429 print(f'Total time: {total_time} seconds')
1430 if path.exists():
1431     print(f'Opening {pdf_path} ...')
1432     os.system(f'open {pdf_path}')
1433 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.) sopranino/
Section_H.ly').readlines()
1434 open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.) sopranino/Section_H.ly',
'w').writelines(part_lines[15:-1])
1435
1436 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
1437     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1438     signature_copy = abjad.mutate(signatures).copy()
1439     staff_copy = abjad.mutate(staff).copy()
1440     part = abjad.Score()
1441     part.insert(0, staff)
1442     part.insert(0, signature_copy)
1443     part_file = abjad.LilyPondFile.new(
1444         part,
1445         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ly'],
1446         )
1447 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1'
1448 pdf_path = f'{directory}/Section_H.pdf'
1449 path = pathlib.Path('Section_H.pdf')
1450 if path.exists():
1451     print(f'Removing {pdf_path} ...')
1452     path.unlink()
1453 time_1 = time.time()
1454 print(f'Persisting {pdf_path} ...')
1455 result = abjad.persist(part_file).as_pdf(pdf_path)
1456 print(result[0])
1457 print(result[1])
1458 print(result[2])
1459 success = result[3]
1460 if success is False:
1461     print('LilyPond failed!')
1462 time_2 = time.time()
1463 total_time = time_2 - time_1
1464 print(f'Total time: {total_time} seconds')
1465 if path.exists():
1466     print(f'Opening {pdf_path} ...')
1467     os.system(f'open {pdf_path}')

```

```

1468 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/
1469 Section_H.ly').readlines()
1470 open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/Section_H.ly',
1471 'w').writelines(part_lines[15:-1])
1472
1473 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
1474     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1475     signature_copy = abjad.mutate(signatures).copy()
1476     staff_copy = abjad.mutate(staff).copy()
1477     part = abjad.Score()
1478     part.insert(0, staff)
1479     part.insert(0, signature_copy)
1480     part_file = abjad.LilyPondFile.new(
1481         part,
1482         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1483 _stylesheets/abjad.ily'],
1484         )
1485     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
1486     pdf_path = f'{directory}/Section_H.pdf'
1487     path = pathlib.Path('Section_H.pdf')
1488     if path.exists():
1489         print(f'Removing {pdf_path} ...')
1490         path.unlink()
1491     time_1 = time.time()
1492     print(f'Persisting {pdf_path} ...')
1493     result = abjad.persist(part_file).as_pdf(pdf_path)
1494     print(result[0])
1495     print(result[1])
1496     print(result[2])
1497     success = result[3]
1498     if success is False:
1499         print('LilyPond failed!')
1500     time_2 = time.time()
1501     total_time = time_2 - time_1
1502     print(f'Total time: {total_time} seconds')
1503     if path.exists():
1504         print(f'Opening {pdf_path} ...')
1505         os.system(f'open {pdf_path}')
1506     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/
1507 Section_H.ly').readlines()
1508     open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/Section_H.ly',
1509 'w').writelines(part_lines[15:-1])
1510
1511 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
1512     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1513     signature_copy = abjad.mutate(signatures).copy()
1514     staff_copy = abjad.mutate(staff).copy()
1515     part = abjad.Score()
1516     part.insert(0, staff)
1517     part.insert(0, signature_copy)
1518     part_file = abjad.LilyPondFile.new(
1519         part,
1520         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1521 _stylesheets/abjad.ily'],
1522         )
1523     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3'
1524     pdf_path = f'{directory}/Section_H.pdf'
1525     path = pathlib.Path('Section_H.pdf')
1526     if path.exists():

```

```

1521     print(f'Removing {pdf_path} ...')
1522     path.unlink()
1523 time_1 = time.time()
1524 print(f'Persisting {pdf_path} ...')
1525 result = abjad.persist(part_file).as_pdf(pdf_path)
1526 print(result[0])
1527 print(result[1])
1528 print(result[2])
1529 success = result[3]
1530 if success is False:
1531     print('LilyPond failed!')
1532 time_2 = time.time()
1533 total_time = time_2 - time_1
1534 print(f'Total time: {total_time} seconds')
1535 if path.exists():
1536     print(f'Opening {pdf_path} ...')
1537     os.system(f'open {pdf_path}')
1538 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/
Section_H.ly').readlines()
1539 open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.) soprano3/Section_H.ly',
'w').writelines(part_lines[15:-1])
1540
1541 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
1542 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1543 signature_copy = abjad.mutate(signatures).copy()
1544 staff_copy = abjad.mutate(staff).copy()
1545 part = abjad.Score()
1546 part.insert(0, staff)
1547 part.insert(0, signature_copy)
1548 part_file = abjad.LilyPondFile.new(
1549     part,
1550     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1551     )
1552 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1'
1553 pdf_path = f'{directory}/Section_H.pdf'
1554 path = pathlib.Path('Section_H.pdf')
1555 if path.exists():
1556     print(f'Removing {pdf_path} ...')
1557     path.unlink()
1558 time_1 = time.time()
1559 print(f'Persisting {pdf_path} ...')
1560 result = abjad.persist(part_file).as_pdf(pdf_path)
1561 print(result[0])
1562 print(result[1])
1563 print(result[2])
1564 success = result[3]
1565 if success is False:
1566     print('LilyPond failed!')
1567 time_2 = time.time()
1568 total_time = time_2 - time_1
1569 print(f'Total time: {total_time} seconds')
1570 if path.exists():
1571     print(f'Opening {pdf_path} ...')
1572     os.system(f'open {pdf_path}')
1573 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/
Section_H.ly').readlines()
1574 open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.) alto1/Section_H.ly',
'w').writelines(part_lines[15:-1])

```

```

1575
1576 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
1577     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1578     signature_copy = abjad.mutate(signatures).copy()
1579     staff_copy = abjad.mutate(staff).copy()
1580     part = abjad.Score()
1581     part.insert(0, staff)
1582     part.insert(0, signature_copy)
1583     part_file = abjad.LilyPondFile.new(
1584         part,
1585         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1586         _stylesheets/abjad.ily'],
1587         )
1588     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
1589     pdf_path = f'{directory}/Section_H.pdf'
1590     path = pathlib.Path('Section_H.pdf')
1591     if path.exists():
1592         print(f'Removing {pdf_path} ...')
1593         path.unlink()
1594     time_1 = time.time()
1595     print(f'Persisting {pdf_path} ...')
1596     result = abjad.persist(part_file).as_pdf(pdf_path)
1597     print(result[0])
1598     print(result[1])
1599     print(result[2])
1600     success = result[3]
1601     if success is False:
1602         print('LilyPond failed!')
1603     time_2 = time.time()
1604     total_time = time_2 - time_1
1605     print(f'Total time: {total_time} seconds')
1606     if path.exists():
1607         print(f'Opening {pdf_path} ...')
1608         os.system(f'open {pdf_path}')
1609     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
1610     Section_H.ly').readlines()
1611     open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_H.ly', 'w'
1612     ).writelines(part_lines[15:-1])

1613
1614 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
1615     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1616     signature_copy = abjad.mutate(signatures).copy()
1617     staff_copy = abjad.mutate(staff).copy()
1618     part = abjad.Score()
1619     part.insert(0, staff)
1620     part.insert(0, signature_copy)
1621     part_file = abjad.LilyPondFile.new(
1622         part,
1623         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1624         _stylesheets/abjad.ily'],
1625         )
1626     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
1627     pdf_path = f'{directory}/Section_H.pdf'
1628     path = pathlib.Path('Section_H.pdf')
1629     if path.exists():
1630         print(f'Removing {pdf_path} ...')
1631         path.unlink()
1632     time_1 = time.time()
1633     print(f'Persisting {pdf_path} ...')

```

```

1630     result = abjad.persist(part_file).as_pdf(pdf_path)
1631     print(result[0])
1632     print(result[1])
1633     print(result[2])
1634     success = result[3]
1635     if success is False:
1636         print('LilyPond failed!')
1637     time_2 = time.time()
1638     total_time = time_2 - time_1
1639     print(f'Total time: {total_time} seconds')
1640     if path.exists():
1641         print(f'Opening {pdf_path} ...')
1642         os.system(f'open {pdf_path}')
1643     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.')alto3/
Section_H.ly').readlines()
1644     open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.')alto3/Section_H.ly', 'w'
).writelines(part_lines[15:-1])
1645
1646 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
1647     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1648     signature_copy = abjad.mutate(signatures).copy()
1649     staff_copy = abjad.mutate(staff).copy()
1650     part = abjad.Score()
1651     part.insert(0, staff)
1652     part.insert(0, signature_copy)
1653     part_file = abjad.LilyPondFile.new(
1654         part,
1655         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
1656         )
1657     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4'
1658     pdf_path = f'{directory}/Section_H.pdf'
1659     path = pathlib.Path('Section_H.pdf')
1660     if path.exists():
1661         print(f'Removing {pdf_path} ...')
1662         path.unlink()
1663     time_1 = time.time()
1664     print(f'Persisting {pdf_path} ...')
1665     result = abjad.persist(part_file).as_pdf(pdf_path)
1666     print(result[0])
1667     print(result[1])
1668     print(result[2])
1669     success = result[3]
1670     if success is False:
1671         print('LilyPond failed!')
1672     time_2 = time.time()
1673     total_time = time_2 - time_1
1674     print(f'Total time: {total_time} seconds')
1675     if path.exists():
1676         print(f'Opening {pdf_path} ...')
1677         os.system(f'open {pdf_path}')
1678     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/
Section_H.ly').readlines()
1679     open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/Section_H.ly', 'w'
).writelines(part_lines[15:-1])
1680
1681 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
1682     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1683     signature_copy = abjad.mutate(signatures).copy()

```

```

1684 staff_copy = abjad.mutate(staff).copy()
1685 part = abjad.Score()
1686 part.insert(0, staff)
1687 part.insert(0, signature_copy)
1688 part_file = abjad.LilyPondFile.new(
1689     part,
1690     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1691 _stylesheets/abjad.ily'],
1692     )
1693 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
1694 pdf_path = f'{directory}/Section_H.pdf'
1695 path = pathlib.Path('Section_H.pdf')
1696 if path.exists():
1697     print(f'Removing {pdf_path} ...')
1698     path.unlink()
1699 time_1 = time.time()
1700 print(f'Persisting {pdf_path} ...')
1701 result = abjad.persist(part_file).as_pdf(pdf_path)
1702 print(result[0])
1703 print(result[1])
1704 print(result[2])
1705 success = result[3]
1706 if success is False:
1707     print('LilyPond failed!')
1708 time_2 = time.time()
1709 total_time = time_2 - time_1
1710 print(f'Total time: {total_time} seconds')
1711 if path.exists():
1712     print(f'Opening {pdf_path} ...')
1713     os.system(f'open {pdf_path}')
1714 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/
1715 Section_H.ly').readlines()
1716 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
1717     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1718     signature_copy = abjad.mutate(signatures).copy()
1719     staff_copy = abjad.mutate(staff).copy()
1720     part = abjad.Score()
1721     part.insert(0, staff)
1722     part.insert(0, signature_copy)
1723     part_file = abjad.LilyPondFile.new(
1724         part,
1725         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1726 _stylesheets/abjad.ily'],
1727         )
1728 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
1729 pdf_path = f'{directory}/Section_H.pdf'
1730 path = pathlib.Path('Section_H.pdf')
1731 if path.exists():
1732     print(f'Removing {pdf_path} ...')
1733     path.unlink()
1734 time_1 = time.time()
1735 print(f'Persisting {pdf_path} ...')
1736 result = abjad.persist(part_file).as_pdf(pdf_path)
1737 print(result[0])
1738 print(result[1])
1739 print(result[2])

```

```

1739     success = result[3]
1740     if success is False:
1741         print('LilyPond failed!')
1742     time_2 = time.time()
1743     total_time = time_2 - time_1
1744     print(f'Total time: {total_time} seconds')
1745     if path.exists():
1746         print(f'Opening {pdf_path} ...')
1747         os.system(f'open {pdf_path}')
1748     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
1749     Section_H.ly').readlines()
1750     open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_H.ly', 'w'
1751     ).writelines(part_lines[15:-1])
1752
1753 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
1754     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1755     signature_copy = abjad.mutate(signatures).copy()
1756     staff_copy = abjad.mutate(staff).copy()
1757     part = abjad.Score()
1758     part.insert(0, staff)
1759     part.insert(0, signature_copy)
1760     part_file = abjad.LilyPondFile.new(
1761         part,
1762         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1763         _stylesheets/abjad.ily'],
1764         )
1765     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
1766     pdf_path = f'{directory}/Section_H.pdf'
1767     path = pathlib.Path('Section_H.pdf')
1768     if path.exists():
1769         print(f'Removing {pdf_path} ...')
1770         path.unlink()
1771     time_1 = time.time()
1772     print(f'Persisting {pdf_path} ...')
1773     result = abjad.persist(part_file).as_pdf(pdf_path)
1774     print(result[0])
1775     print(result[1])
1776     print(result[2])
1777     success = result[3]
1778     if success is False:
1779         print('LilyPond failed!')
1780     time_2 = time.time()
1781     total_time = time_2 - time_1
1782     print(f'Total time: {total_time} seconds')
1783     if path.exists():
1784         print(f'Opening {pdf_path} ...')
1785         os.system(f'open {pdf_path}')
1786     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
1787     Section_H.ly').readlines()
1788     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Section_H.ly', 'w'
1789     ).writelines(part_lines[15:-1])
1790
1791 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
1792     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1793     signature_copy = abjad.mutate(signatures).copy()
1794     staff_copy = abjad.mutate(staff).copy()
1795     part = abjad.Score()
1796     part.insert(0, staff)
1797     part.insert(0, signature_copy)

```

```

1793 part_file = abjad.LilyPondFile.new(
1794     part,
1795     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1796 _stylesheets/abjad.ily'],
1797     )
1798 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
1799 pdf_path = f'{directory}/Section_H.pdf'
1800 path = pathlib.Path('Section_H.pdf')
1801 if path.exists():
1802     print(f'Removing {pdf_path} ...')
1803     path.unlink()
1804 time_1 = time.time()
1805 print(f'Persisting {pdf_path} ...')
1806 result = abjad.persist(part_file).as_pdf(pdf_path)
1807 print(result[0])
1808 print(result[1])
1809 print(result[2])
1810 success = result[3]
1811 if success is False:
1812     print('LilyPond failed!')
1813 time_2 = time.time()
1814 total_time = time_2 - time_1
1815 print(f'Total time: {total_time} seconds')
1816 if path.exists():
1817     print(f'Opening {pdf_path} ...')
1818     os.system(f'open {pdf_path}')
1819 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/
Section_H.ly').readlines()
1820 open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/Section_H.ly', 'w').
writelines(part_lines[15:-1])
1821
1822 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
1823     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1824     signature_copy = abjad.mutate(signatures).copy()
1825     staff_copy = abjad.mutate(staff).copy()
1826     part = abjad.Score()
1827     part.insert(0, staff)
1828     part.insert(0, signature_copy)
1829     part_file = abjad.LilyPondFile.new(
1830         part,
1831         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1832 _stylesheets/abjad.ily'],
1833         )
1834 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3'
1835 pdf_path = f'{directory}/Section_H.pdf'
1836 path = pathlib.Path('Section_H.pdf')
1837 if path.exists():
1838     print(f'Removing {pdf_path} ...')
1839     path.unlink()
1840 time_1 = time.time()
1841 print(f'Persisting {pdf_path} ...')
1842 result = abjad.persist(part_file).as_pdf(pdf_path)
1843 print(result[0])
1844 print(result[1])
1845 print(result[2])
1846 success = result[3]
1847 if success is False:
1848     print('LilyPond failed!')
1849 time_2 = time.time()

```

```

1848     total_time = time_2 - time_1
1849     print(f'Total time: {total_time} seconds')
1850     if path.exists():
1851         print(f'Opening {pdf_path} ...')
1852         os.system(f'open {pdf_path}')
1853     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
1854     Section_H.ly').readlines()
1855     open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_H.ly', ,
1856     w').writelines(part_lines[15:-1])
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
    total_time = time_2 - time_1
    print(f'Total time: {total_time} seconds')
    if path.exists():
        print(f'Opening {pdf_path} ...')
        os.system(f'open {pdf_path}')
    part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
    Section_H.ly').readlines()
    open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_H.ly', ,
    w').writelines(part_lines[15:-1])

for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
    signatures = abjad.select(score['Global Context']).components(abjad.Staff)
    signature_copy = abjad.mutate(signatures).copy()
    staff_copy = abjad.mutate(staff).copy()
    part = abjad.Score()
    part.insert(0, staff)
    part.insert(0, signature_copy)
    part_file = abjad.LilyPondFile.new(
        part,
        includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
        )
    directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'
    pdf_path = f'{directory}/Section_H.pdf'
    path = pathlib.Path('Section_H.pdf')
    if path.exists():
        print(f'Removing {pdf_path} ...')
        path.unlink()
    time_1 = time.time()
    print(f'Persisting {pdf_path} ...')
    result = abjad.persist(part_file).as_pdf(pdf_path)
    print(result[0])
    print(result[1])
    print(result[2])
    success = result[3]
    if success is False:
        print('LilyPond failed!')
    time_2 = time.time()
    total_time = time_2 - time_1
    print(f'Total time: {total_time} seconds')
    if path.exists():
        print(f'Opening {pdf_path} ...')
        os.system(f'open {pdf_path}')
    part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
    Section_H.ly').readlines()
    open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_H.ly', ,
    w').writelines(part_lines[15:-1])

for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
    signatures = abjad.select(score['Global Context']).components(abjad.Staff)
    signature_copy = abjad.mutate(signatures).copy()
    staff_copy = abjad.mutate(staff).copy()
    part = abjad.Score()
    part.insert(0, staff)
    part.insert(0, signature_copy)
    part_file = abjad.LilyPondFile.new(
        part,
        includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
        )

```

```

1901     )
1902     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5'
1903     pdf_path = f'{directory}/Section_H.pdf'
1904     path = pathlib.Path('Section_H.pdf')
1905     if path.exists():
1906         print(f'Removing {pdf_path} ...')
1907         path.unlink()
1908     time_1 = time.time()
1909     print(f'Persisting {pdf_path} ...')
1910     result = abjad.persist(part_file).as_pdf(pdf_path)
1911     print(result[0])
1912     print(result[1])
1913     print(result[2])
1914     success = result[3]
1915     if success is False:
1916         print('LilyPond failed!')
1917     time_2 = time.time()
1918     total_time = time_2 - time_1
1919     print(f'Total time: {total_time} seconds')
1920     if path.exists():
1921         print(f'Opening {pdf_path} ...')
1922         os.system(f'open {pdf_path}')
1923     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/
1924     Section_H.ly').readlines()
1925     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.)tenor5/Section_H.ly', 'w').writelines(part_lines[15:-1])
1926
1927 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
1928     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1929     signature_copy = abjad.mutate(signatures).copy()
1930     staff_copy = abjad.mutate(staff).copy()
1931     part = abjad.Score()
1932     part.insert(0, staff)
1933     part.insert(0, signature_copy)
1934     part_file = abjad.LilyPondFile.new(
1935         part,
1936         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
1937         _stylesheets/abjad.ly'],
1938         )
1939     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1'
1940     pdf_path = f'{directory}/Section_H.pdf'
1941     path = pathlib.Path('Section_H.pdf')
1942     if path.exists():
1943         print(f'Removing {pdf_path} ...')
1944         path.unlink()
1945     time_1 = time.time()
1946     print(f'Persisting {pdf_path} ...')
1947     result = abjad.persist(part_file).as_pdf(pdf_path)
1948     print(result[0])
1949     print(result[1])
1950     print(result[2])
1951     success = result[3]
1952     if success is False:
1953         print('LilyPond failed!')
1954     time_2 = time.time()
1955     total_time = time_2 - time_1
1956     print(f'Total time: {total_time} seconds')
1957     if path.exists():
1958         print(f'Opening {pdf_path} ...')

```

```

1957     os.system(f'open {pdf_path}')
1958     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/
Section_H.ly').readlines()
1959     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/Section_H.ly',
1960     , 'w').writelines(part_lines[15:-1])
1961
1961 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
1962     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1963     signature_copy = abjad.mutate(signatures).copy()
1964     staff_copy = abjad.mutate(staff).copy()
1965     part = abjad.Score()
1966     part.insert(0, staff)
1967     part.insert(0, signature_copy)
1968     part_file = abjad.LilyPondFile.new(
1969         part,
1970         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1971         _stylesheets/abjad.ily'],
1972         )
1972 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2'
1973 pdf_path = f'{directory}/Section_H.pdf'
1974 path = pathlib.Path('Section_H.pdf')
1975 if path.exists():
1976     print(f'Removing {pdf_path} ...')
1977     path.unlink()
1978 time_1 = time.time()
1979 print(f'Persisting {pdf_path} ...')
1980 result = abjad.persist(part_file).as_pdf(pdf_path)
1981 print(result[0])
1982 print(result[1])
1983 print(result[2])
1984 success = result[3]
1985 if success is False:
1986     print('LilyPond failed!')
1987 time_2 = time.time()
1988 total_time = time_2 - time_1
1989 print(f'Total time: {total_time} seconds')
1990 if path.exists():
1991     print(f'Opening {pdf_path} ...')
1992     os.system(f'open {pdf_path}')
1993     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/
Section_H.ly').readlines()
1994     open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/Section_H.ly',
1995     , 'w').writelines(part_lines[15:-1])
1996
1996 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
1997     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1998     signature_copy = abjad.mutate(signatures).copy()
1999     staff_copy = abjad.mutate(staff).copy()
2000     part = abjad.Score()
2001     part.insert(0, staff)
2002     part.insert(0, signature_copy)
2003     part_file = abjad.LilyPondFile.new(
2004         part,
2005         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2006         _stylesheets/abjad.ily'],
2007         )
2007 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.')baritone3'
2008 pdf_path = f'{directory}/Section_H.pdf'
2009 path = pathlib.Path('Section_H.pdf')
```

```

2010     if path.exists():
2011         print(f'Removing {pdf_path} ...')
2012         path.unlink()
2013     time_1 = time.time()
2014     print(f'Persisting {pdf_path} ...')
2015     result = abjad.persist(part_file).as_pdf(pdf_path)
2016     print(result[0])
2017     print(result[1])
2018     print(result[2])
2019     success = result[3]
2020     if success is False:
2021         print('LilyPond failed!')
2022     time_2 = time.time()
2023     total_time = time_2 - time_1
2024     print(f'Total time: {total_time} seconds')
2025     if path.exists():
2026         print(f'Opening {pdf_path} ...')
2027         os.system(f'open {pdf_path}')
2028     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/
2029     Section_H.ly').readlines()
2030     open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/Section_H.ly',
2031     'w').writelines(part_lines[15:-1])
2032
2033 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
2034     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2035     signature_copy = abjad.mutate(signatures).copy()
2036     staff_copy = abjad.mutate(staff).copy()
2037     part = abjad.Score()
2038     part.insert(0, staff)
2039     part.insert(0, signature_copy)
2040     part_file = abjad.LilyPondFile.new(
2041         part,
2042         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2043         _stylesheets/abjad.ily'],
2044         )
2045     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
2046     pdf_path = f'{directory}/Section_H.pdf'
2047     path = pathlib.Path('Section_H.pdf')
2048     if path.exists():
2049         print(f'Removing {pdf_path} ...')
2050         path.unlink()
2051     time_1 = time.time()
2052     print(f'Persisting {pdf_path} ...')
2053     result = abjad.persist(part_file).as_pdf(pdf_path)
2054     print(result[0])
2055     print(result[1])
2056     print(result[2])
2057     success = result[3]
2058     if success is False:
2059         print('LilyPond failed!')
2060     time_2 = time.time()
2061     total_time = time_2 - time_1
2062     print(f'Total time: {total_time} seconds')
2063     if path.exists():
2064         print(f'Opening {pdf_path} ...')
2065         os.system(f'open {pdf_path}')
2066     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
2067     Section_H.ly').readlines()
2068     open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_H.ly', 'w'

```

```

').writelines(part_lines[15:-1])

2065
2066 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
2067     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2068     signature_copy = abjad.mutate(signatures).copy()
2069     staff_copy = abjad.mutate(staff).copy()
2070     part = abjad.Score()
2071     part.insert(0, staff)
2072     part.insert(0, signature_copy)
2073     part_file = abjad.LilyPondFile.new(
2074         part,
2075         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
2076         _stylesheets/abjad.ly'],
2077         )
2078     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
2079     pdf_path = f'{directory}/Section_H.pdf'
2080     path = pathlib.Path('Section_H.pdf')
2081     if path.exists():
2082         print(f'Removing {pdf_path} ...')
2083         path.unlink()
2084     time_1 = time.time()
2085     print(f'Persisting {pdf_path} ...')
2086     result = abjad.persist(part_file).as_pdf(pdf_path)
2087     print(result[0])
2088     print(result[1])
2089     print(result[2])
2090     success = result[3]
2091     if success is False:
2092         print('LilyPond failed!')
2093     time_2 = time.time()
2094     total_time = time_2 - time_1
2095     print(f'Total time: {total_time} seconds')
2096     if path.exists():
2097         print(f'Opening {pdf_path} ...')
2098         os.system(f'open {pdf_path}')
2099     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
2100     Section_H.ly').readlines()
2101     open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_H.ly', 'w'
2102     ).writelines(part_lines[15:-1])

2103
2104 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
2105     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2106     signature_copy = abjad.mutate(signatures).copy()
2107     staff_copy = abjad.mutate(staff).copy()
2108     part = abjad.Score()
2109     part.insert(0, staff)
2110     part.insert(0, signature_copy)
2111     part_file = abjad.LilyPondFile.new(
2112         part,
2113         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
2114         _stylesheets/abjad.ly'],
2115         )
2116     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
2117     pdf_path = f'{directory}/Section_H.pdf'
2118     path = pathlib.Path('Section_H.pdf')
2119     if path.exists():
2120         print(f'Removing {pdf_path} ...')
2121         path.unlink()
2122     time_1 = time.time()

```

```

2119 print(f'Persisting {pdf_path} ...')
2120 result = abjad.persist(part_file).as_pdf(pdf_path)
2121 print(result[0])
2122 print(result[1])
2123 print(result[2])
2124 success = result[3]
2125 if success is False:
2126     print('LilyPond failed!')
2127 time_2 = time.time()
2128 total_time = time_2 - time_1
2129 print(f'Total time: {total_time} seconds')
2130 if path.exists():
2131     print(f'Opening {pdf_path} ...')
2132     os.system(f'open {pdf_path}')
2133 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass
2134 /Section_H.ly').readlines()
2135 open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_H.ly
2136 ', 'w').writelines(part_lines[15:-1])

```

Listing 3.9: Invocation Source Code

## 3.10 Section I

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11
12 print('Interpreting file ...')
13
14 time_signatures = [
15     abjad.TimeSignature(pair) for pair in [
16         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
17         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
18         (4, 4), (4, 4), (4, 4), (4, 4),
19     ]
20 ]
21
22 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
23
24 def cyc(lst):
25     count = 0
26     while True:
27         yield lst[count%len(lst)]
28         count += 1
29
30 def grouper(lst1, lst2):
31     def cyc(lst):
32         c = 0
33         while True:
34             yield lst[c%len(lst)]

```

```

35         c += 1
36     lst1 = cyc(lst1)
37     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2
38 ]
39
40 def reduceMod(list_length, rw):
41     return [(x % list_length) for x in rw]
42
43 # -3 at bottom of chord for completion
44 soprano_chord = [27, ]
45 soprano_1_chord = [13, 16, 22,]
46 soprano_2_chord = [13, 14, 16,]
47 soprano_3_chord = [12, 15, 13,]
48 alto_1_chord = [12, 19, 20,]
49 alto_2_chord = [12, 15, 25, 12,]
50 alto_3_chord = [1, 13, 22, 1,]
51 alto_4_chord = [12, 15, 25, 20,]
52 alto_5_chord = [1, 13, 22, 27, 12,]
53 alto_6_chord = [12, 19, 1,]
54 tenor_1_chord = [6, 17, 17,]
55 tenor_2_chord = [6, 17, 25, 6,]
56 tenor_3_chord = [6, 17, 25, -1]
57 tenor_4_chord = [6, 17, 17,]
58 tenor_5_chord = [6, 17, 25, 6,]
59 baritone_1_chord = [13, 13,]
60 baritone_2_chord = [4, 16, 23, 6,]
61 baritone_3_chord = [7, 17, 25, 4,]
62 bass_1_chord = [11, 9, 9, 11, 9, 11,]
63 bass_2_chord = [9, 11, 11, 9, 11, 9,]
64 contrabass_chord = [-2, 2, 7, -2, 2, 7, 2, -2]
65
66 def reduceMod(x, rw):
67     return [(y % x) for y in rw]
68
69 seed(143)
70 soprano_random_walk = []
71 soprano_random_walk.append(-1 if random() < 0.5 else 1)
72 for i in range(1, 1000):
73     movement = -1 if random() < 0.5 else 1
74     value = soprano_random_walk[i-1] + movement
75     soprano_random_walk.append(value)
76 soprano_walk_chord = [18, 19, 20, 21, 22, 22, 21, 20, 19, 18, 17, 16, 15,
77     14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5,
78     6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ]
79 l = len(soprano_walk_chord)
80 soprano_random_walk_notes = [soprano_walk_chord[x] for x in reduceMod(l,
81     soprano_random_walk)]
82
83 seed(144)
84 soprano_1_random_walk = []
85 soprano_1_random_walk.append(-1 if random() < 0.5 else 1)
86 for i in range(1, 1000):
87     movement = -1 if random() < 0.5 else 1
88     value = soprano_1_random_walk[i-1] + movement
89     soprano_1_random_walk.append(value)
90 soprano_1_walk_chord = [13, 14, 16, 17, 22, 23, ]
91 l = len(soprano_1_walk_chord)
92 soprano_1_random_walk_notes = [soprano_1_walk_chord[x] for x in reduceMod(l,
93     soprano_1_random_walk)]
```

```
89 seed(145)
90 soprano_2_random_walk = []
91 soprano_2_random_walk.append(-1 if random() < 0.5 else 1)
92 for i in range(1, 1000):
93     movement = -1 if random() < 0.5 else 1
94     value = soprano_2_random_walk[i-1] + movement
95     soprano_2_random_walk.append(value)
96 soprano_2_random_walk.append(value)
97 soprano_2_walk_chord = [13, 14, 15, 16, 17, ]
98 l = len(soprano_2_walk_chord)
99 soprano_2_random_walk_notes = [soprano_2_walk_chord[x] for x in reduceMod(l,
100                             soprano_2_random_walk)]
101
102 seed(146)
103 soprano_3_random_walk = []
104 soprano_3_random_walk.append(-1 if random() < 0.5 else 1)
105 for i in range(1, 1000):
106     movement = -1 if random() < 0.5 else 1
107     value = soprano_3_random_walk[i-1] + movement
108     soprano_3_random_walk.append(value)
109 soprano_3_random_walk.append(value)
110 soprano_3_walk_chord = [12, 13, 15, 16, 13, 14, ]
111 l = len(soprano_3_walk_chord)
112 soprano_3_random_walk_notes = [soprano_3_walk_chord[x] for x in reduceMod(l,
113                             soprano_3_random_walk)]
114
115 seed(147)
116 alto_1_random_walk = []
117 alto_1_random_walk.append(-1 if random() < 0.5 else 1)
118 for i in range(1, 1000):
119     movement = -1 if random() < 0.5 else 1
120     value = alto_1_random_walk[i-1] + movement
121     alto_1_random_walk.append(value)
122 alto_1_walk_chord = [12, 13, 19, 20, 21, ]
123 l = len(alto_1_walk_chord)
124 alto_1_random_walk_notes = [alto_1_walk_chord[x] for x in reduceMod(l,
125                             alto_1_random_walk)]
126
127 seed(148)
128 alto_2_random_walk = []
129 alto_2_random_walk.append(-1 if random() < 0.5 else 1)
130 for i in range(1, 1000):
131     movement = -1 if random() < 0.5 else 1
132     value = alto_2_random_walk[i-1] + movement
133     alto_2_random_walk.append(value)
134 alto_2_walk_chord = [12, 13, 15, 16, 25, 26, 12, 13, ]
135 l = len(alto_2_walk_chord)
136 alto_2_random_walk_notes = [alto_2_walk_chord[x] for x in reduceMod(l,
137                             alto_2_random_walk)]
138
139 seed(149)
140 alto_3_random_walk = []
141 alto_3_random_walk.append(-1 if random() < 0.5 else 1)
142 for i in range(1, 1000):
143     movement = -1 if random() < 0.5 else 1
144     value = alto_3_random_walk[i-1] + movement
145     alto_3_random_walk.append(value)
146 alto_3_walk_chord = [1, 2, 13, 14, 22, 23, ]
```

```
144 l = len(alto_3_walk_chord)
145 alto_3_random_walk_notes = [alto_3_walk_chord[x] for x in reduceMod(1,
    alto_3_random_walk)]
146
147 seed(150)
148 alto_4_random_walk = []
149 alto_4_random_walk.append(-1 if random() < 0.5 else 1)
150 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
151     value = alto_4_random_walk[i-1] + movement
152     alto_4_random_walk.append(value)
153 alto_4_walk_chord = [12, 13, 15, 16, 25, 26, 20, 21, ]
154 l = len(alto_4_walk_chord)
155 alto_4_random_walk_notes = [alto_4_walk_chord[x] for x in reduceMod(1,
    alto_4_random_walk)]
156
157 seed(151)
158 alto_5_random_walk = []
159 alto_5_random_walk.append(-1 if random() < 0.5 else 1)
160 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
161     value = alto_5_random_walk[i-1] + movement
162     alto_5_random_walk.append(value)
163 alto_5_walk_chord = [1, 2, 13, 14, 22, 23, 27, 28, 12, 13, ]
164 l = len(alto_5_walk_chord)
165 alto_5_random_walk_notes = [alto_5_walk_chord[x] for x in reduceMod(1,
    alto_5_random_walk)]
166
167 seed(152)
168 alto_6_random_walk = []
169 alto_6_random_walk.append(-1 if random() < 0.5 else 1)
170 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
171     value = alto_6_random_walk[i-1] + movement
172     alto_6_random_walk.append(value)
173 alto_6_walk_chord = [1, 2, 12, 13, 19, 20, ]
174 l = len(alto_6_walk_chord)
175 alto_6_random_walk_notes = [alto_6_walk_chord[x] for x in reduceMod(1,
    alto_6_random_walk)]
176
177 seed(153)
178 tenor_1_random_walk = []
179 tenor_1_random_walk.append(-1 if random() < 0.5 else 1)
180 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
181     value = tenor_1_random_walk[i-1] + movement
182     tenor_1_random_walk.append(value)
183 tenor_1_walk_chord = [6, 7, 18, 17, ]
184 l = len(tenor_1_walk_chord)
185 tenor_1_random_walk_notes = [tenor_1_walk_chord[x] for x in reduceMod(1,
    tenor_1_random_walk)]
186
187 seed(154)
188 tenor_2_random_walk = []
189 tenor_2_random_walk.append(-1 if random() < 0.5 else 1)
190 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
191     value = tenor_2_random_walk[i-1] + movement
192     tenor_2_random_walk.append(value)
```

```

198 tenor_2_walk_chord = [6, 7, 17, 18, 25, 26, ]
199 l = len(tenor_2_walk_chord)
200 tenor_2_random_walk_notes = [tenor_2_walk_chord[x] for x in reduceMod(1,
    tenor_2_random_walk)]
201
202 seed(155)
203 tenor_3_random_walk = []
204 tenor_3_random_walk.append(-1 if random() < 0.5 else 1)
205 for i in range(1, 1000):
206     movement = -1 if random() < 0.5 else 1
207     value = tenor_3_random_walk[i-1] + movement
208     tenor_3_random_walk.append(value)
209 tenor_3_walk_chord = [6, 7, 17, 18, 25, 26, -1, -2, ]
210 l = len(tenor_3_walk_chord)
211 tenor_3_random_walk_notes = [tenor_3_walk_chord[x] for x in reduceMod(1,
    tenor_3_random_walk)]
212
213 seed(156)
214 tenor_4_random_walk = []
215 tenor_4_random_walk.append(-1 if random() < 0.5 else 1)
216 for i in range(1, 1000):
217     movement = -1 if random() < 0.5 else 1
218     value = tenor_4_random_walk[i-1] + movement
219     tenor_4_random_walk.append(value)
220 tenor_4_walk_chord = [6, 7, 17, 18, ]
221 l = len(tenor_4_walk_chord)
222 tenor_4_random_walk_notes = [tenor_4_walk_chord[x] for x in reduceMod(1,
    tenor_4_random_walk)]
223
224 seed(157)
225 tenor_5_random_walk = []
226 tenor_5_random_walk.append(-1 if random() < 0.5 else 1)
227 for i in range(1, 1000):
228     movement = -1 if random() < 0.5 else 1
229     value = tenor_5_random_walk[i-1] + movement
230     tenor_5_random_walk.append(value)
231 tenor_5_walk_chord = [6, 7, 17, 18, 25, 26, ]
232 l = len(tenor_5_walk_chord)
233 tenor_5_random_walk_notes = [tenor_5_walk_chord[x] for x in reduceMod(1,
    tenor_5_random_walk)]
234
235 seed(158)
236 baritone_1_random_walk = []
237 baritone_1_random_walk.append(-1 if random() < 0.5 else 1)
238 for i in range(1, 1000):
239     movement = -1 if random() < 0.5 else 1
240     value = baritone_1_random_walk[i-1] + movement
241     baritone_1_random_walk.append(value)
242 baritone_1_walk_chord = [13, 14, ]
243 l = len(baritone_1_walk_chord)
244 baritone_1_random_walk_notes = [baritone_1_walk_chord[x] for x in reduceMod(1,
    baritone_1_random_walk)]
245
246 seed(159)
247 baritone_2_random_walk = []
248 baritone_2_random_walk.append(-1 if random() < 0.5 else 1)
249 for i in range(1, 1000):
250     movement = -1 if random() < 0.5 else 1
251     value = baritone_2_random_walk[i-1] + movement

```

```

252     baritone_2_random_walk.append(value)
253 baritone_2_walk_chord = [4, 5, 16, 17, 23, 24, 6, 7, ]
254 l = len(baritone_2_walk_chord)
255 baritone_2_random_walk_notes = [baritone_2_walk_chord[x] for x in reduceMod(l,
256                             baritone_2_random_walk)]
257
258 seed(160)
259 baritone_3_random_walk = []
260 baritone_3_random_walk.append(-1 if random() < 0.5 else 1)
261 for i in range(1, 1000):
262     movement = -1 if random() < 0.5 else 1
263     value = baritone_3_random_walk[i-1] + movement
264     baritone_3_random_walk.append(value)
265 baritone_3_walk_chord = [7, 8, 17, 18, 25, 26, 4, 5, ]
266 l = len(baritone_3_walk_chord)
267 baritone_3_random_walk_notes = [baritone_3_walk_chord[x] for x in reduceMod(l,
268                             baritone_3_random_walk)]
269
270 seed(161)
271 bass_1_random_walk = []
272 bass_1_random_walk.append(-1 if random() < 0.5 else 1)
273 for i in range(1, 1000):
274     movement = -1 if random() < 0.5 else 1
275     value = bass_1_random_walk[i-1] + movement
276     bass_1_random_walk.append(value)
277 bass_1_walk_chord = [11, 12, 9, 10, ]
278 l = len(bass_1_walk_chord)
279 bass_1_random_walk_notes = [bass_1_walk_chord[x] for x in reduceMod(l,
280                             bass_1_random_walk)]
281
282 seed(162)
283 bass_2_random_walk = []
284 bass_2_random_walk.append(-1 if random() < 0.5 else 1)
285 for i in range(1, 1000):
286     movement = -1 if random() < 0.5 else 1
287     value = bass_2_random_walk[i-1] + movement
288     bass_2_random_walk.append(value)
289 bass_2_walk_chord = [9, 10, 11, 12, ]
290 l = len(bass_2_walk_chord)
291 bass_2_random_walk_notes = [bass_2_walk_chord[x] for x in reduceMod(l,
292                             bass_2_random_walk)]
293
294 seed(163)
295 contrabass_random_walk = []
296 contrabass_random_walk.append(-1 if random() < 0.5 else 1)
297 for i in range(1, 1000):
298     movement = -1 if random() < 0.5 else 1
299     value = contrabass_random_walk[i-1] + movement
300     contrabass_random_walk.append(value)
301 contrabass_walk_chord = [-2, -1, 2, 3, 7, 8, ]
302 l = len(contrabass_walk_chord)
303 contrabass_random_walk_notes = [contrabass_walk_chord[x] for x in reduceMod(l,
304                             contrabass_random_walk)]
305
306 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
307     talea=abjadext.rmakers.Talea(
308         counts=[3, 5, 4, 6, 2, 7, ],
309         denominator=16,
310     ),

```

```

306     beamSpecifier=abjadext.rmakers.BeamSpecifier(
307         beamDivisionsTogether=True,
308         beamRests=False,
309         ),
310     extraCountsPerDivision=[0, 1, -1, 1, 0, -1, 0, 1,
311     logicalTieMasks=[
312         abjadext.rmakers.silence([2], 5),
313         ],
314     divisionMasks=[
315         abjadext.rmakers.SilenceMask(
316             pattern=abjad.index([1], 7),
317             ),
318         ],
319     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
320         trivialize=True,
321         extractTrivial=True,
322         rewriteRestFilled=True,
323         rewriteSustained=True,
324         ),
325     )
326
327 attachmentHandlerOne = AttachmentHandler(
328     startingDynamic='mf',
329     endingDynamic='mp',
330     hairpin='>',
331 )
332
333 attachmentHandlerTwo = AttachmentHandler(
334     startingDynamic='p',
335     # endingDynamic='pp',
336     hairpin='--',
337 )
338
339 #####sopranino#####
340 sopraninoMusicmakerOne = MusicMaker(
341     rmaker=rmakerOne,
342     pitches=sopraninoRandomWalkNotes,
343     continuous=True,
344     attachmentHandler=attachmentHandlerOne,
345 )
346 sopraninoMusicmakerTwo = MusicMaker(
347     rmaker=rmakerOne,
348     pitches=sopraninoRandomWalkNotes,
349     continuous=True,
350     attachmentHandler=attachmentHandlerTwo,
351 )
352 #####soprano_one#####
353 sopranoOneMusicmakerOne = MusicMaker(
354     rmaker=rmakerOne,
355     pitches=soprano_1RandomWalkNotes,
356     continuous=True,
357     attachmentHandler=attachmentHandlerOne,
358 )
359 sopranoOneMusicmakerTwo = MusicMaker(
360     rmaker=rmakerOne,
361     pitches=soprano_1RandomWalkNotes,
362     continuous=True,
363     attachmentHandler=attachmentHandlerTwo,
364 )

```

```
365 #####soprano_two#####
366 soprano_two_musicmaker_one = MusicMaker(
367     rmaker=rmaker_one,
368     pitches=soprano_2_random_walk_notes,
369     continuous=True,
370     attachment_handler=attachment_handler_one,
371 )
372 soprano_two_musicmaker_two = MusicMaker(
373     rmaker=rmaker_one,
374     pitches=soprano_2_random_walk_notes,
375     continuous=True,
376     attachment_handler=attachment_handler_two,
377 )
378 #####soprano_three#####
379 soprano_three_musicmaker_one = MusicMaker(
380     rmaker=rmaker_one,
381     pitches=soprano_3_random_walk_notes,
382     continuous=True,
383     attachment_handler=attachment_handler_one,
384 )
385 soprano_three_musicmaker_two = MusicMaker(
386     rmaker=rmaker_one,
387     pitches=soprano_3_random_walk_notes,
388     continuous=True,
389     attachment_handler=attachment_handler_two,
390 )
391 #####alto_one#####
392 alto_one_musicmaker_one = MusicMaker(
393     rmaker=rmaker_one,
394     pitches=alto_1_random_walk_notes,
395     continuous=True,
396     attachment_handler=attachment_handler_one,
397 )
398 alto_one_musicmaker_two = MusicMaker(
399     rmaker=rmaker_one,
400     pitches=alto_1_random_walk_notes,
401     continuous=True,
402     attachment_handler=attachment_handler_two,
403 )
404 #####alto_two#####
405 alto_two_musicmaker_one = MusicMaker(
406     rmaker=rmaker_one,
407     pitches=alto_2_random_walk_notes,
408     continuous=True,
409     attachment_handler=attachment_handler_one,
410 )
411 alto_two_musicmaker_two = MusicMaker(
412     rmaker=rmaker_one,
413     pitches=alto_2_random_walk_notes,
414     continuous=True,
415     attachment_handler=attachment_handler_two,
416 )
417 #####alto_three#####
418 alto_three_musicmaker_one = MusicMaker(
419     rmaker=rmaker_one,
420     pitches=alto_3_random_walk_notes,
421     continuous=True,
422     attachment_handler=attachment_handler_one,
423 )
```

```
424 alto_three_musicmaker_two = MusicMaker(
425     rmaker=rmaker_one,
426     pitches=alto_3_random_walk_notes,
427     continuous=True,
428     attachment_handler=attachment_handler_two,
429 )
430 #####alto_four#####
431 alto_four_musicmaker_one = MusicMaker(
432     rmaker=rmaker_one,
433     pitches=alto_4_random_walk_notes,
434     continuous=True,
435     attachment_handler=attachment_handler_one,
436 )
437 alto_four_musicmaker_two = MusicMaker(
438     rmaker=rmaker_one,
439     pitches=alto_4_random_walk_notes,
440     continuous=True,
441     attachment_handler=attachment_handler_two,
442 )
443 #####alto_five#####
444 alto_five_musicmaker_one = MusicMaker(
445     rmaker=rmaker_one,
446     pitches=alto_5_random_walk_notes,
447     continuous=True,
448     attachment_handler=attachment_handler_one,
449 )
450 alto_five_musicmaker_two = MusicMaker(
451     rmaker=rmaker_one,
452     pitches=alto_5_random_walk_notes,
453     continuous=True,
454     attachment_handler=attachment_handler_two,
455 )
456 #####alto_six#####
457 alto_six_musicmaker_one = MusicMaker(
458     rmaker=rmaker_one,
459     pitches=alto_6_random_walk_notes,
460     continuous=True,
461     attachment_handler=attachment_handler_one,
462 )
463 alto_six_musicmaker_two = MusicMaker(
464     rmaker=rmaker_one,
465     pitches=alto_6_random_walk_notes,
466     continuous=True,
467     attachment_handler=attachment_handler_two,
468 )
469 #####tenor_one#####
470 tenor_one_musicmaker_one = MusicMaker(
471     rmaker=rmaker_one,
472     pitches=tenor_1_random_walk_notes,
473     continuous=True,
474     attachment_handler=attachment_handler_one,
475 )
476 tenor_one_musicmaker_two = MusicMaker(
477     rmaker=rmaker_one,
478     pitches=tenor_1_random_walk_notes,
479     continuous=True,
480     attachment_handler=attachment_handler_two,
481 )
482 #####tenor_two#####
```

```
483 tenor_two_musicmaker_one = MusicMaker(  
484     rmaker=rmaker_one,  
485     pitches=tenor_2_random_walk_notes,  
486     continuous=True,  
487     attachment_handler=attachment_handler_one,  
488 )  
489 tenor_two_musicmaker_two = MusicMaker(  
490     rmaker=rmaker_one,  
491     pitches=tenor_2_random_walk_notes,  
492     continuous=True,  
493     attachment_handler=attachment_handler_two,  
494 )  
495 #####tenor_three#####  
496 tenor_three_musicmaker_one = MusicMaker(  
497     rmaker=rmaker_one,  
498     pitches=tenor_3_random_walk_notes,  
499     continuous=True,  
500     attachment_handler=attachment_handler_one,  
501 )  
502 tenor_three_musicmaker_two = MusicMaker(  
503     rmaker=rmaker_one,  
504     pitches=tenor_3_random_walk_notes,  
505     continuous=True,  
506     attachment_handler=attachment_handler_two,  
507 )  
508 #####tenor_four#####  
509 tenor_four_musicmaker_one = MusicMaker(  
510     rmaker=rmaker_one,  
511     pitches=tenor_4_random_walk_notes,  
512     continuous=True,  
513     attachment_handler=attachment_handler_one,  
514 )  
515 tenor_four_musicmaker_two = MusicMaker(  
516     rmaker=rmaker_one,  
517     pitches=tenor_4_random_walk_notes,  
518     continuous=True,  
519     attachment_handler=attachment_handler_two,  
520 )  
521 #####tenor_five#####  
522 tenor_five_musicmaker_one = MusicMaker(  
523     rmaker=rmaker_one,  
524     pitches=tenor_5_random_walk_notes,  
525     continuous=True,  
526     attachment_handler=attachment_handler_one,  
527 )  
528 tenor_five_musicmaker_two = MusicMaker(  
529     rmaker=rmaker_one,  
530     pitches=tenor_5_random_walk_notes,  
531     continuous=True,  
532     attachment_handler=attachment_handler_two,  
533 )  
534 #####baritone_one#####  
535 baritone_one_musicmaker_one = MusicMaker(  
536     rmaker=rmaker_one,  
537     pitches=baritone_1_random_walk_notes,  
538     continuous=True,  
539     attachment_handler=attachment_handler_one,  
540 )  
541 baritone_one_musicmaker_two = MusicMaker(
```

```
542     rmaker=rmaker_one,
543     pitches=baritone_1_random_walk_notes,
544     continuous=True,
545     attachment_handler=attachment_handler_two,
546 )
547 #####baritone_two#####
548 baritone_two_musicmaker_one = MusicMaker(
549     rmaker=rmaker_one,
550     pitches=baritone_2_random_walk_notes,
551     continuous=True,
552     attachment_handler=attachment_handler_one,
553 )
554 baritone_two_musicmaker_two = MusicMaker(
555     rmaker=rmaker_one,
556     pitches=baritone_2_random_walk_notes,
557     continuous=True,
558     attachment_handler=attachment_handler_two,
559 )
560 #####baritone_three#####
561 baritone_three_musicmaker_one = MusicMaker(
562     rmaker=rmaker_one,
563     pitches=baritone_3_random_walk_notes,
564     continuous=True,
565     attachment_handler=attachment_handler_one,
566 )
567 baritone_three_musicmaker_two = MusicMaker(
568     rmaker=rmaker_one,
569     pitches=baritone_3_random_walk_notes,
570     continuous=True,
571     attachment_handler=attachment_handler_two,
572 )
573 #####bass_one#####
574 bass_one_musicmaker_one = MusicMaker(
575     rmaker=rmaker_one,
576     pitches=bass_1_random_walk_notes,
577     continuous=True,
578     attachment_handler=attachment_handler_one,
579 )
580 bass_one_musicmaker_two = MusicMaker(
581     rmaker=rmaker_one,
582     pitches=bass_1_random_walk_notes,
583     continuous=True,
584     attachment_handler=attachment_handler_two,
585 )
586 #####bass_two#####
587 bass_two_musicmaker_one = MusicMaker(
588     rmaker=rmaker_one,
589     pitches=bass_2_random_walk_notes,
590     continuous=True,
591     attachment_handler=attachment_handler_one,
592 )
593 bass_two_musicmaker_two = MusicMaker(
594     rmaker=rmaker_one,
595     pitches=bass_2_random_walk_notes,
596     continuous=True,
597     attachment_handler=attachment_handler_two,
598 )
599 #####contrabass#####
600 contrabass_musicmaker_one = MusicMaker(
```

```

601     rmaker=rmaker_one,
602     pitches=contrabass_random_walk_notes,
603     continuous=True,
604     attachment_handler=attachment_handler_one,
605 )
606 contrabass_musicmaker_two = MusicMaker(
607     rmaker=rmaker_one,
608     pitches=contrabass_random_walk_notes,
609     continuous=True,
610     attachment_handler=attachment_handler_two,
611 )
612
613 silence_maker = abjadext.rmakers.NoteRhythmMaker(
614     division_masks=[
615         abjadext.rmakers.SilenceMask(
616             pattern=abjad.index([0], 1),
617             ),
618         ],
619     )
620
621 class MusicSpecifier:
622
623     def __init__(self, music_maker, voice_name):
624         self.music_maker = music_maker
625         self.voice_name = voice_name
626
627 print('Collecting timespans and rmakers ...')
628
629 voice_1_timespan_list = abjad.TimespanList([
630     abjad.AnnotatedTimespan(
631         start_offset=start_offset,
632         stop_offset=stop_offset,
633         annotation=MusicSpecifier(
634             music_maker=music_maker,
635             voice_name='Voice 1',
636         ),
637     )
638     for start_offset, stop_offset, music_maker in [
639         [(0, 8), (2, 8), sopranino_musicmaker_two],
640         [(2, 8), (4, 8), sopranino_musicmaker_one],
641         [(4, 8), (6, 8), sopranino_musicmaker_two],
642         [(6, 8), (8, 8), sopranino_musicmaker_two],
643
644         [(8, 8), (10, 8), sopranino_musicmaker_one],
645         [(10, 8), (12, 8), sopranino_musicmaker_two],
646         [(12, 8), (14, 8), sopranino_musicmaker_two],
647         [(14, 8), (16, 8), sopranino_musicmaker_two],
648
649         [(16, 8), (18, 8), sopranino_musicmaker_one],
650         [(18, 8), (20, 8), sopranino_musicmaker_two],
651         [(20, 8), (22, 8), sopranino_musicmaker_two],
652         [(22, 8), (24, 8), sopranino_musicmaker_two],
653
654         [(24, 8), (26, 8), sopranino_musicmaker_two],
655         [(26, 8), (28, 8), sopranino_musicmaker_one],
656         [(28, 8), (30, 8), sopranino_musicmaker_two],
657         [(30, 8), (32, 8), sopranino_musicmaker_two],
658
659         [(32, 8), (34, 8), sopranino_musicmaker_two],

```

```

660     [(34, 8), (36, 8), sopranino_musicmaker_two],
661     [(36, 8), (38, 8), sopranino_musicmaker_one],
662     [(38, 8), (40, 8), sopranino_musicmaker_one],
663
664     [(40, 8), (42, 8), sopranino_musicmaker_two],
665     [(42, 8), (44, 8), sopranino_musicmaker_two],
666     [(44, 8), (46, 8), sopranino_musicmaker_two],
667     [(46, 8), (48, 8), sopranino_musicmaker_one],
668
669     [(48, 8), (50, 8), sopranino_musicmaker_one],
670     [(50, 8), (52, 8), sopranino_musicmaker_one],
671     [(52, 8), (54, 8), sopranino_musicmaker_two],
672     [(54, 8), (56, 8), sopranino_musicmaker_two],
673
674     [(56, 8), (58, 8), sopranino_musicmaker_one],
675     [(58, 8), (60, 8), sopranino_musicmaker_one],
676     [(60, 8), (62, 8), sopranino_musicmaker_one],
677     [(62, 8), (64, 8), sopranino_musicmaker_two],
678
679     [(64, 8), (66, 8), sopranino_musicmaker_two],
680     [(66, 8), (68, 8), sopranino_musicmaker_one],
681     [(68, 8), (70, 8), sopranino_musicmaker_one],
682     [(70, 8), (72, 8), sopranino_musicmaker_two],
683
684     [(72, 8), (74, 8), sopranino_musicmaker_one],
685     [(74, 8), (76, 8), sopranino_musicmaker_one],
686     [(76, 8), (78, 8), sopranino_musicmaker_two],
687     [(78, 8), (80, 8), sopranino_musicmaker_one],
688
689     [(80, 8), (82, 8), sopranino_musicmaker_one],
690     [(82, 8), (84, 8), sopranino_musicmaker_one],
691     [(84, 8), (86, 8), sopranino_musicmaker_one],
692     [(86, 8), (88, 8), sopranino_musicmaker_two],
693
694     [(88, 8), (90, 8), sopranino_musicmaker_two],
695     [(90, 8), (92, 8), sopranino_musicmaker_two],
696     [(92, 8), (94, 8), sopranino_musicmaker_two],
697     [(94, 8), (96, 8), sopranino_musicmaker_one],
698
699     [(96, 8), (98, 8), sopranino_musicmaker_one],
700     [(98, 8), (100, 8), sopranino_musicmaker_one],
701     [(100, 8), (102, 8), sopranino_musicmaker_one],
702     [(102, 8), (104, 8), sopranino_musicmaker_two],
703
704     [(104, 8), (106, 8), sopranino_musicmaker_two],
705     [(106, 8), (108, 8), sopranino_musicmaker_two],
706     [(108, 8), (110, 8), sopranino_musicmaker_one],
707     [(110, 8), (112, 8), sopranino_musicmaker_two],
708 ]
709 ])
710
711 voice_2_timespan_list = abjad.TimespanList([
712     abjad.AnnotatedTimespan(
713         start_offset=start_offset,
714         stop_offset=stop_offset,
715         annotation=MusicSpecifier(
716             music_maker=music_maker,
717             voice_name='Voice 2',
718         ),

```

```
719     )
720     for start_offset, stop_offset, music_maker in [
721         [(0, 8), (2, 8), soprano_one_musicmaker_two],
722         [(2, 8), (4, 8), soprano_one_musicmaker_two],
723         [(4, 8), (6, 8), soprano_one_musicmaker_two],
724         [(6, 8), (8, 8), soprano_one_musicmaker_two],
725
726         [(8, 8), (10, 8), soprano_one_musicmaker_two],
727         [(10, 8), (12, 8), soprano_one_musicmaker_one],
728         [(12, 8), (14, 8), soprano_one_musicmaker_two],
729         [(14, 8), (16, 8), soprano_one_musicmaker_one],
730
731         [(16, 8), (18, 8), soprano_one_musicmaker_one],
732         [(18, 8), (20, 8), soprano_one_musicmaker_one],
733         [(20, 8), (22, 8), soprano_one_musicmaker_two],
734         [(22, 8), (24, 8), soprano_one_musicmaker_one],
735
736         [(24, 8), (26, 8), soprano_one_musicmaker_one],
737         [(26, 8), (28, 8), soprano_one_musicmaker_one],
738         [(28, 8), (30, 8), soprano_one_musicmaker_two],
739         [(30, 8), (32, 8), soprano_one_musicmaker_two],
740
741         [(32, 8), (34, 8), soprano_one_musicmaker_two],
742         [(34, 8), (36, 8), soprano_one_musicmaker_two],
743         [(36, 8), (38, 8), soprano_one_musicmaker_two],
744         [(38, 8), (40, 8), soprano_one_musicmaker_one],
745
746         [(40, 8), (42, 8), soprano_one_musicmaker_one],
747         [(42, 8), (44, 8), soprano_one_musicmaker_one],
748         [(44, 8), (46, 8), soprano_one_musicmaker_one],
749         [(46, 8), (48, 8), soprano_one_musicmaker_two],
750
751         [(48, 8), (50, 8), soprano_one_musicmaker_two],
752         [(50, 8), (52, 8), soprano_one_musicmaker_two],
753         [(52, 8), (54, 8), soprano_one_musicmaker_two],
754         [(54, 8), (56, 8), soprano_one_musicmaker_one],
755
756         [(56, 8), (58, 8), soprano_one_musicmaker_one],
757         [(58, 8), (60, 8), soprano_one_musicmaker_one],
758         [(60, 8), (62, 8), soprano_one_musicmaker_two],
759         [(62, 8), (64, 8), soprano_one_musicmaker_one],
760
761         [(64, 8), (66, 8), soprano_one_musicmaker_one],
762         [(66, 8), (68, 8), soprano_one_musicmaker_one],
763         [(68, 8), (70, 8), soprano_one_musicmaker_two],
764         [(70, 8), (72, 8), soprano_one_musicmaker_two],
765
766         [(72, 8), (74, 8), soprano_one_musicmaker_two],
767         [(74, 8), (76, 8), soprano_one_musicmaker_two],
768         [(76, 8), (78, 8), soprano_one_musicmaker_one],
769         [(78, 8), (80, 8), soprano_one_musicmaker_one],
770
771         [(80, 8), (82, 8), soprano_one_musicmaker_one],
772         [(82, 8), (84, 8), soprano_one_musicmaker_two],
773         [(84, 8), (86, 8), soprano_one_musicmaker_two],
774         [(86, 8), (88, 8), soprano_one_musicmaker_one],
775
776         [(88, 8), (90, 8), soprano_one_musicmaker_two],
777         [(90, 8), (92, 8), soprano_one_musicmaker_two],
```

```

778     [(92, 8), (94, 8), soprano_one_musicmaker_two],
779     [(94, 8), (96, 8), soprano_one_musicmaker_one],
780
781     [(96, 8), (98, 8), soprano_one_musicmaker_one],
782     [(98, 8), (100, 8), soprano_one_musicmaker_one],
783     [(100, 8), (102, 8), soprano_one_musicmaker_two],
784     [(102, 8), (104, 8), soprano_one_musicmaker_two],
785
786     [(104, 8), (106, 8), soprano_one_musicmaker_two],
787     [(106, 8), (108, 8), soprano_one_musicmaker_two],
788     [(108, 8), (110, 8), soprano_one_musicmaker_one],
789     [(110, 8), (112, 8), soprano_one_musicmaker_one],
790 ]
791 ])
792
793 voice_3_timespan_list = abjad.TimespanList([
794     abjad.AnnotatedTimespan(
795         start_offset=start_offset,
796         stop_offset=stop_offset,
797         annotation=MusicSpecifier(
798             music_maker=music_maker,
799             voice_name='Voice 3',
800         ),
801     )
802     for start_offset, stop_offset, music_maker in [
803         [(0, 8), (2, 8), soprano_two_musicmaker_one],
804         [(2, 8), (4, 8), soprano_two_musicmaker_one],
805         [(4, 8), (6, 8), soprano_two_musicmaker_two],
806         [(6, 8), (8, 8), soprano_two_musicmaker_two],
807
808         [(8, 8), (10, 8), soprano_two_musicmaker_one],
809         [(10, 8), (12, 8), soprano_two_musicmaker_one],
810         [(12, 8), (14, 8), soprano_two_musicmaker_two],
811         [(14, 8), (16, 8), soprano_two_musicmaker_one],
812
813         [(16, 8), (18, 8), soprano_two_musicmaker_two],
814         [(18, 8), (20, 8), soprano_two_musicmaker_one],
815         [(20, 8), (22, 8), soprano_two_musicmaker_two],
816         [(22, 8), (24, 8), soprano_two_musicmaker_two],
817
818         [(24, 8), (26, 8), soprano_two_musicmaker_one],
819         [(26, 8), (28, 8), soprano_two_musicmaker_one],
820         [(28, 8), (30, 8), soprano_two_musicmaker_two],
821         [(30, 8), (32, 8), soprano_two_musicmaker_one],
822
823         [(32, 8), (34, 8), soprano_two_musicmaker_one],
824         [(34, 8), (36, 8), soprano_two_musicmaker_one],
825         [(36, 8), (38, 8), soprano_two_musicmaker_one],
826         [(38, 8), (40, 8), soprano_two_musicmaker_two],
827
828         [(40, 8), (42, 8), soprano_two_musicmaker_one],
829         [(42, 8), (44, 8), soprano_two_musicmaker_two],
830         [(44, 8), (46, 8), soprano_two_musicmaker_two],
831         [(46, 8), (48, 8), soprano_two_musicmaker_one],
832
833         [(48, 8), (50, 8), soprano_two_musicmaker_two],
834         [(50, 8), (52, 8), soprano_two_musicmaker_one],
835         [(52, 8), (54, 8), soprano_two_musicmaker_one],
836         [(54, 8), (56, 8), soprano_two_musicmaker_two],

```

```

837
838     [(56, 8), (58, 8), soprano_two_musicmaker_one],
839     [(58, 8), (60, 8), soprano_two_musicmaker_two],
840     [(60, 8), (62, 8), soprano_two_musicmaker_one],
841     [(62, 8), (64, 8), soprano_two_musicmaker_two],
842
843     [(64, 8), (66, 8), soprano_two_musicmaker_one],
844     [(66, 8), (68, 8), soprano_two_musicmaker_one],
845     [(68, 8), (70, 8), soprano_two_musicmaker_two],
846     [(70, 8), (72, 8), soprano_two_musicmaker_one],
847
848     [(72, 8), (74, 8), soprano_two_musicmaker_two],
849     [(74, 8), (76, 8), soprano_two_musicmaker_two],
850     [(76, 8), (78, 8), soprano_two_musicmaker_one],
851     [(78, 8), (80, 8), soprano_two_musicmaker_one],
852
853     [(80, 8), (82, 8), soprano_two_musicmaker_one],
854     [(82, 8), (84, 8), soprano_two_musicmaker_two],
855     [(84, 8), (86, 8), soprano_two_musicmaker_two],
856     [(86, 8), (88, 8), soprano_two_musicmaker_two],
857
858     [(88, 8), (90, 8), soprano_two_musicmaker_two],
859     [(90, 8), (92, 8), soprano_two_musicmaker_one],
860     [(92, 8), (94, 8), soprano_two_musicmaker_one],
861     [(94, 8), (96, 8), soprano_two_musicmaker_one],
862
863     [(96, 8), (98, 8), soprano_two_musicmaker_one],
864     [(98, 8), (100, 8), soprano_two_musicmaker_two],
865     [(100, 8), (102, 8), soprano_two_musicmaker_two],
866     [(102, 8), (104, 8), soprano_two_musicmaker_two],
867
868     [(104, 8), (106, 8), soprano_two_musicmaker_two],
869     [(106, 8), (108, 8), soprano_two_musicmaker_one],
870     [(108, 8), (110, 8), soprano_two_musicmaker_one],
871     [(110, 8), (112, 8), soprano_two_musicmaker_one],
872 ]
873 ])
874
875 voice_4_timespan_list = abjad.TimespanList([
876     abjad.AnnotatedTimespan(
877         start_offset=start_offset,
878         stop_offset=stop_offset,
879         annotation=MusicSpecifier(
880             music_maker=music_maker,
881             voice_name='Voice 4',
882         ),
883     )
884     for start_offset, stop_offset, music_maker in [
885         [(0, 8), (2, 8), soprano_three_musicmaker_one],
886         [(2, 8), (4, 8), soprano_three_musicmaker_two],
887         [(4, 8), (6, 8), soprano_three_musicmaker_two],
888         [(6, 8), (8, 8), soprano_three_musicmaker_two],
889
890         [(8, 8), (10, 8), soprano_three_musicmaker_one],
891         [(10, 8), (12, 8), soprano_three_musicmaker_one],
892         [(12, 8), (14, 8), soprano_three_musicmaker_one],
893         [(14, 8), (16, 8), soprano_three_musicmaker_one],
894
895         [(16, 8), (18, 8), soprano_three_musicmaker_two],

```

```

896     [(18, 8), (20, 8), soprano_three_musicmaker_two],
897     [(20, 8), (22, 8), soprano_three_musicmaker_two],
898     [(22, 8), (24, 8), soprano_three_musicmaker_two],
899
900     [(24, 8), (26, 8), soprano_three_musicmaker_two],
901     [(26, 8), (28, 8), soprano_three_musicmaker_one],
902     [(28, 8), (30, 8), soprano_three_musicmaker_two],
903     [(30, 8), (32, 8), soprano_three_musicmaker_one],
904
905     [(32, 8), (34, 8), soprano_three_musicmaker_one],
906     [(34, 8), (36, 8), soprano_three_musicmaker_two],
907     [(36, 8), (38, 8), soprano_three_musicmaker_two],
908     [(38, 8), (40, 8), soprano_three_musicmaker_one],
909
910     [(40, 8), (42, 8), soprano_three_musicmaker_one],
911     [(42, 8), (44, 8), soprano_three_musicmaker_one],
912     [(44, 8), (46, 8), soprano_three_musicmaker_two],
913     [(46, 8), (48, 8), soprano_three_musicmaker_two],
914
915     [(48, 8), (50, 8), soprano_three_musicmaker_two],
916     [(50, 8), (52, 8), soprano_three_musicmaker_two],
917     [(52, 8), (54, 8), soprano_three_musicmaker_two],
918     [(54, 8), (56, 8), soprano_three_musicmaker_one],
919
920     [(56, 8), (58, 8), soprano_three_musicmaker_two],
921     [(58, 8), (60, 8), soprano_three_musicmaker_one],
922     [(60, 8), (62, 8), soprano_three_musicmaker_one],
923     [(62, 8), (64, 8), soprano_three_musicmaker_one],
924
925     [(64, 8), (66, 8), soprano_three_musicmaker_one],
926     [(66, 8), (68, 8), soprano_three_musicmaker_one],
927     [(68, 8), (70, 8), soprano_three_musicmaker_two],
928     [(70, 8), (72, 8), soprano_three_musicmaker_one],
929
930     [(72, 8), (74, 8), soprano_three_musicmaker_two],
931     [(74, 8), (76, 8), soprano_three_musicmaker_one],
932     [(76, 8), (78, 8), soprano_three_musicmaker_two],
933     [(78, 8), (80, 8), soprano_three_musicmaker_two],
934
935     [(80, 8), (82, 8), soprano_three_musicmaker_two],
936     [(82, 8), (84, 8), soprano_three_musicmaker_two],
937     [(84, 8), (86, 8), soprano_three_musicmaker_one],
938     [(86, 8), (88, 8), soprano_three_musicmaker_one],
939
940     [(88, 8), (90, 8), soprano_three_musicmaker_one],
941     [(90, 8), (92, 8), soprano_three_musicmaker_one],
942     [(92, 8), (94, 8), soprano_three_musicmaker_one],
943     [(94, 8), (96, 8), soprano_three_musicmaker_two],
944
945     [(96, 8), (98, 8), soprano_three_musicmaker_one],
946     [(98, 8), (100, 8), soprano_three_musicmaker_one],
947     [(100, 8), (102, 8), soprano_three_musicmaker_two],
948     [(102, 8), (104, 8), soprano_three_musicmaker_one],
949
950     [(104, 8), (106, 8), soprano_three_musicmaker_one],
951     [(106, 8), (108, 8), soprano_three_musicmaker_two],
952     [(108, 8), (110, 8), soprano_three_musicmaker_two],
953     [(110, 8), (112, 8), soprano_three_musicmaker_two],
954 ]

```

```
955 ])
956
957 voice_5_timespan_list = abjad.TimespanList([
958     abjad.AnnotatedTimespan(
959         start_offset=start_offset,
960         stop_offset=stop_offset,
961         annotation=MusicSpecifier(
962             music_maker=music_maker,
963             voice_name='Voice 5',
964         ),
965     )
966     for start_offset, stop_offset, music_maker in [
967         [(0, 8), (2, 8), alto_one_musicmaker_one],
968         [(2, 8), (4, 8), alto_one_musicmaker_one],
969         [(4, 8), (6, 8), alto_one_musicmaker_two],
970         [(6, 8), (8, 8), alto_one_musicmaker_two],
971
972         [(8, 8), (10, 8), alto_one_musicmaker_two],
973         [(10, 8), (12, 8), alto_one_musicmaker_one],
974         [(12, 8), (14, 8), alto_one_musicmaker_two],
975         [(14, 8), (16, 8), alto_one_musicmaker_one],
976
977         [(16, 8), (18, 8), alto_one_musicmaker_one],
978         [(18, 8), (20, 8), alto_one_musicmaker_one],
979         [(20, 8), (22, 8), alto_one_musicmaker_one],
980         [(22, 8), (24, 8), alto_one_musicmaker_one],
981
982         [(24, 8), (26, 8), alto_one_musicmaker_two],
983         [(26, 8), (28, 8), alto_one_musicmaker_two],
984         [(28, 8), (30, 8), alto_one_musicmaker_two],
985         [(30, 8), (32, 8), alto_one_musicmaker_two],
986
987         [(32, 8), (34, 8), alto_one_musicmaker_two],
988         [(34, 8), (36, 8), alto_one_musicmaker_one],
989         [(36, 8), (38, 8), alto_one_musicmaker_one],
990         [(38, 8), (40, 8), alto_one_musicmaker_one],
991
992         [(40, 8), (42, 8), alto_one_musicmaker_two],
993         [(42, 8), (44, 8), alto_one_musicmaker_one],
994         [(44, 8), (46, 8), alto_one_musicmaker_two],
995         [(46, 8), (48, 8), alto_one_musicmaker_two],
996
997         [(48, 8), (50, 8), alto_one_musicmaker_two],
998         [(50, 8), (52, 8), alto_one_musicmaker_one],
999         [(52, 8), (54, 8), alto_one_musicmaker_two],
1000         [(54, 8), (56, 8), alto_one_musicmaker_one],
1001
1002         [(56, 8), (58, 8), alto_one_musicmaker_one],
1003         [(58, 8), (60, 8), alto_one_musicmaker_one],
1004         [(60, 8), (62, 8), alto_one_musicmaker_two],
1005         [(62, 8), (64, 8), alto_one_musicmaker_one],
1006
1007         [(64, 8), (66, 8), alto_one_musicmaker_two],
1008         [(66, 8), (68, 8), alto_one_musicmaker_one],
1009         [(68, 8), (70, 8), alto_one_musicmaker_one],
1010         [(70, 8), (72, 8), alto_one_musicmaker_two],
1011
1012         [(72, 8), (74, 8), alto_one_musicmaker_two],
1013         [(74, 8), (76, 8), alto_one_musicmaker_one],
```

```

1014     [(76, 8), (78, 8), alto_one_musicmaker_one],
1015     [(78, 8), (80, 8), alto_one_musicmaker_one],
1016
1017     [(80, 8), (82, 8), alto_one_musicmaker_one],
1018     [(82, 8), (84, 8), alto_one_musicmaker_two],
1019     [(84, 8), (86, 8), alto_one_musicmaker_two],
1020     [(86, 8), (88, 8), alto_one_musicmaker_two],
1021
1022     [(88, 8), (90, 8), alto_one_musicmaker_two],
1023     [(90, 8), (92, 8), alto_one_musicmaker_one],
1024     [(92, 8), (94, 8), alto_one_musicmaker_two],
1025     [(94, 8), (96, 8), alto_one_musicmaker_one],
1026
1027     [(96, 8), (98, 8), alto_one_musicmaker_one],
1028     [(98, 8), (100, 8), alto_one_musicmaker_two],
1029     [(100, 8), (102, 8), alto_one_musicmaker_two],
1030     [(102, 8), (104, 8), alto_one_musicmaker_one],
1031
1032     [(104, 8), (106, 8), alto_one_musicmaker_one],
1033     [(106, 8), (108, 8), alto_one_musicmaker_one],
1034     [(108, 8), (110, 8), alto_one_musicmaker_two],
1035     [(110, 8), (112, 8), alto_one_musicmaker_two],
1036 ]
1037 ])
1038
1039 voice_6_timespan_list = abjad.TimespanList([
1040     abjad.AnnotatedTimespan(
1041         start_offset=start_offset,
1042         stop_offset=stop_offset,
1043         annotation=MusicSpecifier(
1044             music_maker=music_maker,
1045             voice_name='Voice 6',
1046         ),
1047     )
1048     for start_offset, stop_offset, music_maker in [
1049         [(0, 8), (2, 8), alto_two_musicmaker_one],
1050         [(2, 8), (4, 8), alto_two_musicmaker_one],
1051         [(4, 8), (6, 8), alto_two_musicmaker_one],
1052         [(6, 8), (8, 8), alto_two_musicmaker_one],
1053
1054         [(8, 8), (10, 8), alto_two_musicmaker_two],
1055         [(10, 8), (12, 8), alto_two_musicmaker_two],
1056         [(12, 8), (14, 8), alto_two_musicmaker_two],
1057         [(14, 8), (16, 8), alto_two_musicmaker_two],
1058
1059         [(16, 8), (18, 8), alto_two_musicmaker_two],
1060         [(18, 8), (20, 8), alto_two_musicmaker_one],
1061         [(20, 8), (22, 8), alto_two_musicmaker_one],
1062         [(22, 8), (24, 8), alto_two_musicmaker_one],
1063
1064         [(24, 8), (26, 8), alto_two_musicmaker_two],
1065         [(26, 8), (28, 8), alto_two_musicmaker_one],
1066         [(28, 8), (30, 8), alto_two_musicmaker_one],
1067         [(30, 8), (32, 8), alto_two_musicmaker_two],
1068
1069         [(32, 8), (34, 8), alto_two_musicmaker_two],
1070         [(34, 8), (36, 8), alto_two_musicmaker_two],
1071         [(36, 8), (38, 8), alto_two_musicmaker_two],
1072         [(38, 8), (40, 8), alto_two_musicmaker_one],

```

```

1073     [(40, 8), (42, 8), alto_two_musicmaker_one],
1074     [(42, 8), (44, 8), alto_two_musicmaker_one],
1075     [(44, 8), (46, 8), alto_two_musicmaker_two],
1076     [(46, 8), (48, 8), alto_two_musicmaker_two],
1077
1078     [(48, 8), (50, 8), alto_two_musicmaker_one],
1079     [(50, 8), (52, 8), alto_two_musicmaker_two],
1080     [(52, 8), (54, 8), alto_two_musicmaker_one],
1081     [(54, 8), (56, 8), alto_two_musicmaker_two],
1082
1083     [(56, 8), (58, 8), alto_two_musicmaker_two],
1084     [(58, 8), (60, 8), alto_two_musicmaker_one],
1085     [(60, 8), (62, 8), alto_two_musicmaker_one],
1086     [(62, 8), (64, 8), alto_two_musicmaker_two],
1087
1088     [(64, 8), (66, 8), alto_two_musicmaker_two],
1089     [(66, 8), (68, 8), alto_two_musicmaker_two],
1090     [(68, 8), (70, 8), alto_two_musicmaker_one],
1091     [(70, 8), (72, 8), alto_two_musicmaker_one],
1092
1093     [(72, 8), (74, 8), alto_two_musicmaker_one],
1094     [(74, 8), (76, 8), alto_two_musicmaker_two],
1095     [(76, 8), (78, 8), alto_two_musicmaker_two],
1096     [(78, 8), (80, 8), alto_two_musicmaker_two],
1097
1098     [(80, 8), (82, 8), alto_two_musicmaker_two],
1099     [(82, 8), (84, 8), alto_two_musicmaker_one],
1100     [(84, 8), (86, 8), alto_two_musicmaker_one],
1101     [(86, 8), (88, 8), alto_two_musicmaker_one],
1102
1103     [(88, 8), (90, 8), alto_two_musicmaker_one],
1104     [(90, 8), (92, 8), alto_two_musicmaker_two],
1105     [(92, 8), (94, 8), alto_two_musicmaker_two],
1106     [(94, 8), (96, 8), alto_two_musicmaker_one],
1107
1108     [(96, 8), (98, 8), alto_two_musicmaker_two],
1109     [(98, 8), (100, 8), alto_two_musicmaker_two],
1110     [(100, 8), (102, 8), alto_two_musicmaker_one],
1111     [(102, 8), (104, 8), alto_two_musicmaker_two],
1112
1113     [(104, 8), (106, 8), alto_two_musicmaker_one],
1114     [(106, 8), (108, 8), alto_two_musicmaker_two],
1115     [(108, 8), (110, 8), alto_two_musicmaker_one],
1116     [(110, 8), (112, 8), alto_two_musicmaker_two],
1117
1118 ]
1119 ])
1120
1121 voice_7_timespan_list = abjad.TimespanList([
1122     abjad.AnnotatedTimespan(
1123         start_offset=start_offset,
1124         stop_offset=stop_offset,
1125         annotation=MusicSpecifier(
1126             music_maker=music_maker,
1127             voice_name='Voice 7',
1128         ),
1129     )
1130     for start_offset, stop_offset, music_maker in [
1131         [(0, 8), (2, 8), alto_three_musicmaker_two],

```

```
1132      [(2, 8), (4, 8), alto_three_musicmaker_two],  
1133      [(4, 8), (6, 8), alto_three_musicmaker_one],  
1134      [(6, 8), (8, 8), alto_three_musicmaker_two],  
1135  
1136      [(8, 8), (10, 8), alto_three_musicmaker_two],  
1137      [(10, 8), (12, 8), alto_three_musicmaker_one],  
1138      [(12, 8), (14, 8), alto_three_musicmaker_two],  
1139      [(14, 8), (16, 8), alto_three_musicmaker_one],  
1140  
1141      [(16, 8), (18, 8), alto_three_musicmaker_one],  
1142      [(18, 8), (20, 8), alto_three_musicmaker_one],  
1143      [(20, 8), (22, 8), alto_three_musicmaker_two],  
1144      [(22, 8), (24, 8), alto_three_musicmaker_two],  
1145  
1146      [(24, 8), (26, 8), alto_three_musicmaker_two],  
1147      [(26, 8), (28, 8), alto_three_musicmaker_one],  
1148      [(28, 8), (30, 8), alto_three_musicmaker_two],  
1149      [(30, 8), (32, 8), alto_three_musicmaker_one],  
1150  
1151      [(32, 8), (34, 8), alto_three_musicmaker_one],  
1152      [(34, 8), (36, 8), alto_three_musicmaker_one],  
1153      [(36, 8), (38, 8), alto_three_musicmaker_two],  
1154      [(38, 8), (40, 8), alto_three_musicmaker_one],  
1155  
1156      [(40, 8), (42, 8), alto_three_musicmaker_one],  
1157      [(42, 8), (44, 8), alto_three_musicmaker_two],  
1158      [(44, 8), (46, 8), alto_three_musicmaker_one],  
1159      [(46, 8), (48, 8), alto_three_musicmaker_one],  
1160  
1161      [(48, 8), (50, 8), alto_three_musicmaker_one],  
1162      [(50, 8), (52, 8), alto_three_musicmaker_two],  
1163      [(52, 8), (54, 8), alto_three_musicmaker_one],  
1164      [(54, 8), (56, 8), alto_three_musicmaker_two],  
1165  
1166      [(56, 8), (58, 8), alto_three_musicmaker_two],  
1167      [(58, 8), (60, 8), alto_three_musicmaker_one],  
1168      [(60, 8), (62, 8), alto_three_musicmaker_one],  
1169      [(62, 8), (64, 8), alto_three_musicmaker_one],  
1170  
1171      [(64, 8), (66, 8), alto_three_musicmaker_two],  
1172      [(66, 8), (68, 8), alto_three_musicmaker_two],  
1173      [(68, 8), (70, 8), alto_three_musicmaker_two],  
1174      [(70, 8), (72, 8), alto_three_musicmaker_two],  
1175  
1176      [(72, 8), (74, 8), alto_three_musicmaker_two],  
1177      [(74, 8), (76, 8), alto_three_musicmaker_one],  
1178      [(76, 8), (78, 8), alto_three_musicmaker_one],  
1179      [(78, 8), (80, 8), alto_three_musicmaker_one],  
1180  
1181      [(80, 8), (82, 8), alto_three_musicmaker_one],  
1182      [(82, 8), (84, 8), alto_three_musicmaker_two],  
1183      [(84, 8), (86, 8), alto_three_musicmaker_one],  
1184      [(86, 8), (88, 8), alto_three_musicmaker_two],  
1185  
1186      [(88, 8), (90, 8), alto_three_musicmaker_two],  
1187      [(90, 8), (92, 8), alto_three_musicmaker_two],  
1188      [(92, 8), (94, 8), alto_three_musicmaker_one],  
1189      [(94, 8), (96, 8), alto_three_musicmaker_two],  
1190
```

```

1191     [(96, 8), (98, 8), alto_three_musicmaker_two],
1192     [(98, 8), (100, 8), alto_three_musicmaker_one],
1193     [(100, 8), (102, 8), alto_three_musicmaker_one],
1194     [(102, 8), (104, 8), alto_three_musicmaker_one],
1195
1196     [(104, 8), (106, 8), alto_three_musicmaker_two],
1197     [(106, 8), (108, 8), alto_three_musicmaker_one],
1198     [(108, 8), (110, 8), alto_three_musicmaker_two],
1199     [(110, 8), (112, 8), alto_three_musicmaker_two],
1200 ]
1201 ])
1202
1203 voice_8_timespan_list = abjad.TimespanList([
1204     abjad.AnnotatedTimespan(
1205         start_offset=start_offset,
1206         stop_offset=stop_offset,
1207         annotation=MusicSpecifier(
1208             music_maker=music_maker,
1209             voice_name='Voice 8',
1210         ),
1211     )
1212     for start_offset, stop_offset, music_maker in [
1213         [(0, 8), (2, 8), alto_four_musicmaker_two],
1214         [(2, 8), (4, 8), alto_four_musicmaker_two],
1215         [(4, 8), (6, 8), alto_four_musicmaker_two],
1216         [(6, 8), (8, 8), alto_four_musicmaker_one],
1217
1218         [(8, 8), (10, 8), alto_four_musicmaker_one],
1219         [(10, 8), (12, 8), alto_four_musicmaker_one],
1220         [(12, 8), (14, 8), alto_four_musicmaker_two],
1221         [(14, 8), (16, 8), alto_four_musicmaker_two],
1222
1223         [(16, 8), (18, 8), alto_four_musicmaker_two],
1224         [(18, 8), (20, 8), alto_four_musicmaker_one],
1225         [(20, 8), (22, 8), alto_four_musicmaker_one],
1226         [(22, 8), (24, 8), alto_four_musicmaker_one],
1227
1228         [(24, 8), (26, 8), alto_four_musicmaker_one],
1229         [(26, 8), (28, 8), alto_four_musicmaker_two],
1230         [(28, 8), (30, 8), alto_four_musicmaker_two],
1231         [(30, 8), (32, 8), alto_four_musicmaker_two],
1232
1233         [(32, 8), (34, 8), alto_four_musicmaker_two],
1234         [(34, 8), (36, 8), alto_four_musicmaker_one],
1235         [(36, 8), (38, 8), alto_four_musicmaker_one],
1236         [(38, 8), (40, 8), alto_four_musicmaker_one],
1237
1238         [(40, 8), (42, 8), alto_four_musicmaker_one],
1239         [(42, 8), (44, 8), alto_four_musicmaker_one],
1240         [(44, 8), (46, 8), alto_four_musicmaker_two],
1241         [(46, 8), (48, 8), alto_four_musicmaker_two],
1242
1243         [(48, 8), (50, 8), alto_four_musicmaker_two],
1244         [(50, 8), (52, 8), alto_four_musicmaker_two],
1245         [(52, 8), (54, 8), alto_four_musicmaker_one],
1246         [(54, 8), (56, 8), alto_four_musicmaker_one],
1247
1248         [(56, 8), (58, 8), alto_four_musicmaker_one],
1249         [(58, 8), (60, 8), alto_four_musicmaker_two],

```

```

1250     [(60, 8), (62, 8), alto_four_musicmaker_one],
1251     [(62, 8), (64, 8), alto_four_musicmaker_two],
1252
1253     [(64, 8), (66, 8), alto_four_musicmaker_one],
1254     [(66, 8), (68, 8), alto_four_musicmaker_two],
1255     [(68, 8), (70, 8), alto_four_musicmaker_one],
1256     [(70, 8), (72, 8), alto_four_musicmaker_two],
1257
1258     [(72, 8), (74, 8), alto_four_musicmaker_two],
1259     [(74, 8), (76, 8), alto_four_musicmaker_two],
1260     [(76, 8), (78, 8), alto_four_musicmaker_one],
1261     [(78, 8), (80, 8), alto_four_musicmaker_one],
1262
1263     [(80, 8), (82, 8), alto_four_musicmaker_one],
1264     [(82, 8), (84, 8), alto_four_musicmaker_two],
1265     [(84, 8), (86, 8), alto_four_musicmaker_one],
1266     [(86, 8), (88, 8), alto_four_musicmaker_two],
1267
1268     [(88, 8), (90, 8), alto_four_musicmaker_one],
1269     [(90, 8), (92, 8), alto_four_musicmaker_one],
1270     [(92, 8), (94, 8), alto_four_musicmaker_two],
1271     [(94, 8), (96, 8), alto_four_musicmaker_two],
1272
1273     [(96, 8), (98, 8), alto_four_musicmaker_two],
1274     [(98, 8), (100, 8), alto_four_musicmaker_two],
1275     [(100, 8), (102, 8), alto_four_musicmaker_two],
1276     [(102, 8), (104, 8), alto_four_musicmaker_one],
1277
1278     [(104, 8), (106, 8), alto_four_musicmaker_two],
1279     [(106, 8), (108, 8), alto_four_musicmaker_one],
1280     [(108, 8), (110, 8), alto_four_musicmaker_two],
1281     [(110, 8), (112, 8), alto_four_musicmaker_one],
1282 ]
1283 ])
1284
1285 voice_9_timespan_list = abjad.TimespanList([
1286     abjad.AnnotatedTimespan(
1287         start_offset=start_offset,
1288         stop_offset=stop_offset,
1289         annotation=MusicSpecifier(
1290             music_maker=music_maker,
1291             voice_name='Voice 9',
1292         ),
1293     )
1294     for start_offset, stop_offset, music_maker in [
1295         [(0, 8), (2, 8), alto_five_musicmaker_two],
1296         [(2, 8), (4, 8), alto_five_musicmaker_one],
1297         [(4, 8), (6, 8), alto_five_musicmaker_one],
1298         [(6, 8), (8, 8), alto_five_musicmaker_two],
1299
1300         [(8, 8), (10, 8), alto_five_musicmaker_two],
1301         [(10, 8), (12, 8), alto_five_musicmaker_two],
1302         [(12, 8), (14, 8), alto_five_musicmaker_one],
1303         [(14, 8), (16, 8), alto_five_musicmaker_two],
1304
1305         [(16, 8), (18, 8), alto_five_musicmaker_one],
1306         [(18, 8), (20, 8), alto_five_musicmaker_one],
1307         [(20, 8), (22, 8), alto_five_musicmaker_two],
1308         [(22, 8), (24, 8), alto_five_musicmaker_one],

```

```

1309 [(24, 8), (26, 8), alto_five_musicmaker_two],
1310 [(26, 8), (28, 8), alto_five_musicmaker_two],
1311 [(28, 8), (30, 8), alto_five_musicmaker_one],
1312 [(30, 8), (32, 8), alto_five_musicmaker_two],
1313
1314 [(32, 8), (34, 8), alto_five_musicmaker_one],
1315 [(34, 8), (36, 8), alto_five_musicmaker_one],
1316 [(36, 8), (38, 8), alto_five_musicmaker_two],
1317 [(38, 8), (40, 8), alto_five_musicmaker_one],
1318
1319 [(40, 8), (42, 8), alto_five_musicmaker_one],
1320 [(42, 8), (44, 8), alto_five_musicmaker_two],
1321 [(44, 8), (46, 8), alto_five_musicmaker_one],
1322 [(46, 8), (48, 8), alto_five_musicmaker_two],
1323
1324 [(48, 8), (50, 8), alto_five_musicmaker_two],
1325 [(50, 8), (52, 8), alto_five_musicmaker_two],
1326 [(52, 8), (54, 8), alto_five_musicmaker_one],
1327 [(54, 8), (56, 8), alto_five_musicmaker_two],
1328
1329 [(56, 8), (58, 8), alto_five_musicmaker_one],
1330 [(58, 8), (60, 8), alto_five_musicmaker_one],
1331 [(60, 8), (62, 8), alto_five_musicmaker_one],
1332 [(62, 8), (64, 8), alto_five_musicmaker_two],
1333
1334 [(64, 8), (66, 8), alto_five_musicmaker_two],
1335 [(66, 8), (68, 8), alto_five_musicmaker_one],
1336 [(68, 8), (70, 8), alto_five_musicmaker_one],
1337 [(70, 8), (72, 8), alto_five_musicmaker_one],
1338
1339 [(72, 8), (74, 8), alto_five_musicmaker_two],
1340 [(74, 8), (76, 8), alto_five_musicmaker_two],
1341 [(76, 8), (78, 8), alto_five_musicmaker_one],
1342 [(78, 8), (80, 8), alto_five_musicmaker_one],
1343
1344 [(80, 8), (82, 8), alto_five_musicmaker_two],
1345 [(82, 8), (84, 8), alto_five_musicmaker_one],
1346 [(84, 8), (86, 8), alto_five_musicmaker_two],
1347 [(86, 8), (88, 8), alto_five_musicmaker_one],
1348
1349 [(88, 8), (90, 8), alto_five_musicmaker_two],
1350 [(90, 8), (92, 8), alto_five_musicmaker_one],
1351 [(92, 8), (94, 8), alto_five_musicmaker_two],
1352 [(94, 8), (96, 8), alto_five_musicmaker_one],
1353
1354 [(96, 8), (98, 8), alto_five_musicmaker_one],
1355 [(98, 8), (100, 8), alto_five_musicmaker_one],
1356 [(100, 8), (102, 8), alto_five_musicmaker_two],
1357 [(102, 8), (104, 8), alto_five_musicmaker_two],
1358
1359 [(104, 8), (106, 8), alto_five_musicmaker_two],
1360 [(106, 8), (108, 8), alto_five_musicmaker_one],
1361 [(108, 8), (110, 8), alto_five_musicmaker_two],
1362 [(110, 8), (112, 8), alto_five_musicmaker_one],
1363
1364 ]
1365 ])
1366
1367 voice_10_timespan_list = abjad.TimespanList([

```

```

1368     abjad.AnnotatedTimespan(
1369         start_offset=start_offset,
1370         stop_offset=stop_offset,
1371         annotation=MusicSpecifier(
1372             music_maker=music_maker,
1373             voice_name='Voice 10',
1374         ),
1375     )
1376     for start_offset, stop_offset, music_maker in [
1377         [(0, 8), (2, 8), alto_six_musicmaker_one],
1378         [(2, 8), (4, 8), alto_six_musicmaker_one],
1379         [(4, 8), (6, 8), alto_six_musicmaker_two],
1380         [(6, 8), (8, 8), alto_six_musicmaker_one],
1381
1382         [(8, 8), (10, 8), alto_six_musicmaker_two],
1383         [(10, 8), (12, 8), alto_six_musicmaker_two],
1384         [(12, 8), (14, 8), alto_six_musicmaker_one],
1385         [(14, 8), (16, 8), alto_six_musicmaker_two],
1386
1387         [(16, 8), (18, 8), alto_six_musicmaker_one],
1388         [(18, 8), (20, 8), alto_six_musicmaker_two],
1389         [(20, 8), (22, 8), alto_six_musicmaker_one],
1390         [(22, 8), (24, 8), alto_six_musicmaker_two],
1391
1392         [(24, 8), (26, 8), alto_six_musicmaker_one],
1393         [(26, 8), (28, 8), alto_six_musicmaker_two],
1394         [(28, 8), (30, 8), alto_six_musicmaker_one],
1395         [(30, 8), (32, 8), alto_six_musicmaker_two],
1396
1397         [(32, 8), (34, 8), alto_six_musicmaker_two],
1398         [(34, 8), (36, 8), alto_six_musicmaker_two],
1399         [(36, 8), (38, 8), alto_six_musicmaker_one],
1400         [(38, 8), (40, 8), alto_six_musicmaker_one],
1401
1402         [(40, 8), (42, 8), alto_six_musicmaker_one],
1403         [(42, 8), (44, 8), alto_six_musicmaker_two],
1404         [(44, 8), (46, 8), alto_six_musicmaker_one],
1405         [(46, 8), (48, 8), alto_six_musicmaker_one],
1406
1407         [(48, 8), (50, 8), alto_six_musicmaker_one],
1408         [(50, 8), (52, 8), alto_six_musicmaker_two],
1409         [(52, 8), (54, 8), alto_six_musicmaker_two],
1410         [(54, 8), (56, 8), alto_six_musicmaker_two],
1411
1412         [(56, 8), (58, 8), alto_six_musicmaker_two],
1413         [(58, 8), (60, 8), alto_six_musicmaker_two],
1414         [(60, 8), (62, 8), alto_six_musicmaker_one],
1415         [(62, 8), (64, 8), alto_six_musicmaker_two],
1416
1417         [(64, 8), (66, 8), alto_six_musicmaker_two],
1418         [(66, 8), (68, 8), alto_six_musicmaker_one],
1419         [(68, 8), (70, 8), alto_six_musicmaker_two],
1420         [(70, 8), (72, 8), alto_six_musicmaker_one],
1421
1422         [(72, 8), (74, 8), alto_six_musicmaker_one],
1423         [(74, 8), (76, 8), alto_six_musicmaker_two],
1424         [(76, 8), (78, 8), alto_six_musicmaker_one],
1425         [(78, 8), (80, 8), alto_six_musicmaker_two],
1426

```

```

1427     [(80, 8), (82, 8), alto_six_musicmaker_two],
1428     [(82, 8), (84, 8), alto_six_musicmaker_two],
1429     [(84, 8), (86, 8), alto_six_musicmaker_one],
1430     [(86, 8), (88, 8), alto_six_musicmaker_two],
1431
1432     [(88, 8), (90, 8), alto_six_musicmaker_one],
1433     [(90, 8), (92, 8), alto_six_musicmaker_one],
1434     [(92, 8), (94, 8), alto_six_musicmaker_two],
1435     [(94, 8), (96, 8), alto_six_musicmaker_one],
1436
1437     [(96, 8), (98, 8), alto_six_musicmaker_two],
1438     [(98, 8), (100, 8), alto_six_musicmaker_two],
1439     [(100, 8), (102, 8), alto_six_musicmaker_two],
1440     [(102, 8), (104, 8), alto_six_musicmaker_one],
1441
1442     [(104, 8), (106, 8), alto_six_musicmaker_one],
1443     [(106, 8), (108, 8), alto_six_musicmaker_one],
1444     [(108, 8), (110, 8), alto_six_musicmaker_two],
1445     [(110, 8), (112, 8), alto_six_musicmaker_two],
1446 ]
1447 ])
1448
1449 voice_11_timespan_list = abjad.TimespanList([
1450     abjad.AnnotatedTimespan(
1451         start_offset=start_offset,
1452         stop_offset=stop_offset,
1453         annotation=MusicSpecifier(
1454             music_maker=music_maker,
1455             voice_name='Voice 11',
1456         ),
1457     )
1458     for start_offset, stop_offset, music_maker in [
1459         [(0, 8), (2, 8), tenor_one_musicmaker_one],
1460         [(2, 8), (4, 8), tenor_one_musicmaker_one],
1461         [(4, 8), (6, 8), tenor_one_musicmaker_two],
1462         [(6, 8), (8, 8), tenor_one_musicmaker_two],
1463         [(8, 8), (10, 8), tenor_one_musicmaker_two],
1464         [(10, 8), (12, 8), tenor_one_musicmaker_one],
1465         [(12, 8), (14, 8), tenor_one_musicmaker_one],
1466         [(14, 8), (16, 8), tenor_one_musicmaker_one],
1467         [(16, 8), (18, 8), tenor_one_musicmaker_one],
1468         [(18, 8), (20, 8), tenor_one_musicmaker_two],
1469         [(20, 8), (22, 8), tenor_one_musicmaker_two],
1470         [(22, 8), (24, 8), tenor_one_musicmaker_two],
1471         [(24, 8), (26, 8), tenor_one_musicmaker_two],
1472         [(26, 8), (28, 8), tenor_one_musicmaker_one],
1473         [(28, 8), (30, 8), tenor_one_musicmaker_one],
1474         [(30, 8), (32, 8), tenor_one_musicmaker_one],
1475         [(32, 8), (34, 8), tenor_one_musicmaker_one],
1476         [(34, 8), (36, 8), tenor_one_musicmaker_one],
1477         [(36, 8), (38, 8), tenor_one_musicmaker_two],
1478         [(38, 8), (40, 8), tenor_one_musicmaker_two],
1479         [(40, 8), (42, 8), tenor_one_musicmaker_two],
1480         [(42, 8), (44, 8), tenor_one_musicmaker_two],
1481         [(44, 8), (46, 8), tenor_one_musicmaker_one],
1482         [(46, 8), (48, 8), tenor_one_musicmaker_one],
1483         [(48, 8), (50, 8), tenor_one_musicmaker_one],
1484         [(50, 8), (52, 8), tenor_one_musicmaker_one],
1485         [(52, 8), (54, 8), tenor_one_musicmaker_two],

```

```

1486     [(54, 8), (56, 8), tenor_one_musicmaker_two],
1487     [(56, 8), (58, 8), tenor_one_musicmaker_two],
1488     [(58, 8), (60, 8), tenor_one_musicmaker_one],
1489     [(60, 8), (62, 8), tenor_one_musicmaker_one],
1490     [(62, 8), (64, 8), tenor_one_musicmaker_one],
1491     [(64, 8), (66, 8), tenor_one_musicmaker_two],
1492     [(66, 8), (68, 8), tenor_one_musicmaker_two],
1493     [(68, 8), (70, 8), tenor_one_musicmaker_one],
1494     [(70, 8), (72, 8), tenor_one_musicmaker_two],
1495     [(72, 8), (74, 8), tenor_one_musicmaker_two],
1496     [(74, 8), (76, 8), tenor_one_musicmaker_two],
1497     [(76, 8), (78, 8), tenor_one_musicmaker_one],
1498     [(78, 8), (80, 8), tenor_one_musicmaker_one],
1499     [(80, 8), (82, 8), tenor_one_musicmaker_one],
1500     [(82, 8), (84, 8), tenor_one_musicmaker_one],
1501     [(84, 8), (86, 8), tenor_one_musicmaker_two],
1502     [(86, 8), (88, 8), tenor_one_musicmaker_two],
1503     [(88, 8), (90, 8), tenor_one_musicmaker_two],
1504     [(90, 8), (92, 8), tenor_one_musicmaker_two],
1505     [(92, 8), (94, 8), tenor_one_musicmaker_two],
1506     [(94, 8), (96, 8), tenor_one_musicmaker_two],
1507     [(96, 8), (98, 8), tenor_one_musicmaker_two],
1508     [(98, 8), (100, 8), tenor_one_musicmaker_one],
1509     [(100, 8), (102, 8), tenor_one_musicmaker_one],
1510     [(102, 8), (104, 8), tenor_one_musicmaker_one],
1511     [(104, 8), (106, 8), tenor_one_musicmaker_one],
1512     [(106, 8), (108, 8), tenor_one_musicmaker_one],
1513     [(108, 8), (110, 8), tenor_one_musicmaker_one],
1514     [(110, 8), (112, 8), tenor_one_musicmaker_one],
1515 ]
1516 ])
1517
1518 voice_12_timestrap_list = abjad.TimespanList([
1519     abjad.AnnotatedTimespan(
1520         start_offset=start_offset,
1521         stop_offset=stop_offset,
1522         annotation=MusicSpecifier(
1523             music_maker=music_maker,
1524             voice_name='Voice 12',
1525         ),
1526     )
1527     for start_offset, stop_offset, music_maker in [
1528         [(0, 8), (2, 8), tenor_two_musicmaker_two],
1529         [(2, 8), (4, 8), tenor_two_musicmaker_one],
1530         [(4, 8), (6, 8), tenor_two_musicmaker_one],
1531         [(6, 8), (8, 8), tenor_two_musicmaker_one],
1532         [(8, 8), (10, 8), tenor_two_musicmaker_one],
1533         [(10, 8), (12, 8), tenor_two_musicmaker_one],
1534         [(12, 8), (14, 8), tenor_two_musicmaker_one],
1535         [(14, 8), (16, 8), tenor_two_musicmaker_two],
1536         [(16, 8), (18, 8), tenor_two_musicmaker_two],
1537         [(18, 8), (20, 8), tenor_two_musicmaker_two],
1538         [(20, 8), (22, 8), tenor_two_musicmaker_two],
1539         [(22, 8), (24, 8), tenor_two_musicmaker_two],
1540         [(24, 8), (26, 8), tenor_two_musicmaker_two],
1541         [(26, 8), (28, 8), tenor_two_musicmaker_two],
1542         [(28, 8), (30, 8), tenor_two_musicmaker_one],
1543         [(30, 8), (32, 8), tenor_two_musicmaker_one],
1544         [(32, 8), (34, 8), tenor_two_musicmaker_one],

```

```

1545     [(34, 8), (36, 8), tenor_two_musicmaker_one],
1546     [(36, 8), (38, 8), tenor_two_musicmaker_one],
1547     [(38, 8), (40, 8), tenor_two_musicmaker_one],
1548     [(40, 8), (42, 8), tenor_two_musicmaker_one],
1549     [(42, 8), (44, 8), tenor_two_musicmaker_two],
1550     [(44, 8), (46, 8), tenor_two_musicmaker_two],
1551     [(46, 8), (48, 8), tenor_two_musicmaker_two],
1552     [(48, 8), (50, 8), tenor_two_musicmaker_one],
1553     [(50, 8), (52, 8), tenor_two_musicmaker_one],
1554     [(52, 8), (54, 8), tenor_two_musicmaker_two],
1555     [(54, 8), (56, 8), tenor_two_musicmaker_two],
1556     [(56, 8), (58, 8), tenor_two_musicmaker_two],
1557     [(58, 8), (60, 8), tenor_two_musicmaker_two],
1558     [(60, 8), (62, 8), tenor_two_musicmaker_two],
1559     [(62, 8), (64, 8), tenor_two_musicmaker_one],
1560     [(64, 8), (66, 8), tenor_two_musicmaker_one],
1561     [(66, 8), (68, 8), tenor_two_musicmaker_one],
1562     [(68, 8), (70, 8), tenor_two_musicmaker_one],
1563     [(70, 8), (72, 8), tenor_two_musicmaker_one],
1564     [(72, 8), (74, 8), tenor_two_musicmaker_two],
1565     [(74, 8), (76, 8), tenor_two_musicmaker_two],
1566     [(76, 8), (78, 8), tenor_two_musicmaker_two],
1567     [(78, 8), (80, 8), tenor_two_musicmaker_two],
1568     [(80, 8), (82, 8), tenor_two_musicmaker_two],
1569     [(82, 8), (84, 8), tenor_two_musicmaker_two],
1570     [(84, 8), (86, 8), tenor_two_musicmaker_two],
1571     [(86, 8), (88, 8), tenor_two_musicmaker_one],
1572     [(88, 8), (90, 8), tenor_two_musicmaker_one],
1573     [(90, 8), (92, 8), tenor_two_musicmaker_one],
1574     [(92, 8), (94, 8), tenor_two_musicmaker_one],
1575     [(94, 8), (96, 8), tenor_two_musicmaker_one],
1576     [(96, 8), (98, 8), tenor_two_musicmaker_one],
1577     [(98, 8), (100, 8), tenor_two_musicmaker_two],
1578     [(100, 8), (102, 8), tenor_two_musicmaker_two],
1579     [(102, 8), (104, 8), tenor_two_musicmaker_two],
1580     [(104, 8), (106, 8), tenor_two_musicmaker_two],
1581     [(106, 8), (108, 8), tenor_two_musicmaker_one],
1582     [(108, 8), (110, 8), tenor_two_musicmaker_one],
1583     [(110, 8), (112, 8), tenor_two_musicmaker_one],
1584   ]
1585 ])
1586
1587 voice_13_timespan_list = abjad.TimespanList([
1588     abjad.AnnotatedTimespan(
1589         start_offset=start_offset,
1590         stop_offset=stop_offset,
1591         annotation=MusicSpecifier(
1592             music_maker=music_maker,
1593             voice_name='Voice 13',
1594         ),
1595     )
1596     for start_offset, stop_offset, music_maker in [
1597         [(0, 8), (2, 8), tenor_three_musicmaker_one],
1598         [(2, 8), (4, 8), tenor_three_musicmaker_one],
1599         [(4, 8), (6, 8), tenor_three_musicmaker_one],
1600         [(6, 8), (8, 8), tenor_three_musicmaker_one],
1601         [(8, 8), (10, 8), tenor_three_musicmaker_one],
1602         [(10, 8), (12, 8), tenor_three_musicmaker_one],
1603         [(12, 8), (14, 8), tenor_three_musicmaker_two],

```

```

1604     [(14, 8), (16, 8), tenor_three_musicmaker_two],
1605     [(16, 8), (18, 8), tenor_three_musicmaker_two],
1606     [(18, 8), (20, 8), tenor_three_musicmaker_two],
1607     [(20, 8), (22, 8), tenor_three_musicmaker_two],
1608     [(22, 8), (24, 8), tenor_three_musicmaker_two],
1609     [(24, 8), (26, 8), tenor_three_musicmaker_one],
1610     [(26, 8), (28, 8), tenor_three_musicmaker_one],
1611     [(28, 8), (30, 8), tenor_three_musicmaker_one],
1612     [(30, 8), (32, 8), tenor_three_musicmaker_two],
1613     [(32, 8), (34, 8), tenor_three_musicmaker_two],
1614     [(34, 8), (36, 8), tenor_three_musicmaker_two],
1615     [(36, 8), (38, 8), tenor_three_musicmaker_two],
1616     [(38, 8), (40, 8), tenor_three_musicmaker_one],
1617     [(40, 8), (42, 8), tenor_three_musicmaker_one],
1618     [(42, 8), (44, 8), tenor_three_musicmaker_one],
1619     [(44, 8), (46, 8), tenor_three_musicmaker_one],
1620     [(46, 8), (48, 8), tenor_three_musicmaker_one],
1621     [(48, 8), (50, 8), tenor_three_musicmaker_one],
1622     [(50, 8), (52, 8), tenor_three_musicmaker_one],
1623     [(52, 8), (54, 8), tenor_three_musicmaker_two],
1624     [(54, 8), (56, 8), tenor_three_musicmaker_two],
1625     [(56, 8), (58, 8), tenor_three_musicmaker_two],
1626     [(58, 8), (60, 8), tenor_three_musicmaker_two],
1627     [(60, 8), (62, 8), tenor_three_musicmaker_two],
1628     [(62, 8), (64, 8), tenor_three_musicmaker_one],
1629     [(64, 8), (66, 8), tenor_three_musicmaker_one],
1630     [(66, 8), (68, 8), tenor_three_musicmaker_one],
1631     [(68, 8), (70, 8), tenor_three_musicmaker_one],
1632     [(70, 8), (72, 8), tenor_three_musicmaker_one],
1633     [(72, 8), (74, 8), tenor_three_musicmaker_two],
1634     [(74, 8), (76, 8), tenor_three_musicmaker_two],
1635     [(76, 8), (78, 8), tenor_three_musicmaker_two],
1636     [(78, 8), (80, 8), tenor_three_musicmaker_two],
1637     [(80, 8), (82, 8), tenor_three_musicmaker_two],
1638     [(82, 8), (84, 8), tenor_three_musicmaker_two],
1639     [(84, 8), (86, 8), tenor_three_musicmaker_two],
1640     [(86, 8), (88, 8), tenor_three_musicmaker_one],
1641     [(88, 8), (90, 8), tenor_three_musicmaker_one],
1642     [(90, 8), (92, 8), tenor_three_musicmaker_one],
1643     [(92, 8), (94, 8), tenor_three_musicmaker_one],
1644     [(94, 8), (96, 8), tenor_three_musicmaker_two],
1645     [(96, 8), (98, 8), tenor_three_musicmaker_two],
1646     [(98, 8), (100, 8), tenor_three_musicmaker_two],
1647     [(100, 8), (102, 8), tenor_three_musicmaker_two],
1648     [(102, 8), (104, 8), tenor_three_musicmaker_two],
1649     [(104, 8), (106, 8), tenor_three_musicmaker_one],
1650     [(106, 8), (108, 8), tenor_three_musicmaker_one],
1651     [(108, 8), (110, 8), tenor_three_musicmaker_one],
1652     [(110, 8), (112, 8), tenor_three_musicmaker_one],
1653 ]
1654 ])
1655
1656 voice_14_timespan_list = abjad.TimespanList([
1657     abjad.AnnotatedTimespan(
1658         start_offset=start_offset,
1659         stop_offset=stop_offset,
1660         annotation=MusicSpecifier(
1661             music_maker=music_maker,
1662             voice_name='Voice 14',

```

```
1663     ),
1664 )
1665 for start_offset, stop_offset, music_maker in [
1666     [(0, 8), (2, 8), tenor_four_musicmaker_two],
1667     [(2, 8), (4, 8), tenor_four_musicmaker_two],
1668     [(4, 8), (6, 8), tenor_four_musicmaker_two],
1669     [(6, 8), (8, 8), tenor_four_musicmaker_two],
1670     [(8, 8), (10, 8), tenor_four_musicmaker_two],
1671     [(10, 8), (12, 8), tenor_four_musicmaker_two],
1672     [(12, 8), (14, 8), tenor_four_musicmaker_two],
1673     [(14, 8), (16, 8), tenor_four_musicmaker_two],
1674     [(16, 8), (18, 8), tenor_four_musicmaker_one],
1675     [(18, 8), (20, 8), tenor_four_musicmaker_one],
1676     [(20, 8), (22, 8), tenor_four_musicmaker_one],
1677     [(22, 8), (24, 8), tenor_four_musicmaker_one],
1678     [(24, 8), (26, 8), tenor_four_musicmaker_one],
1679     [(26, 8), (28, 8), tenor_four_musicmaker_one],
1680     [(28, 8), (30, 8), tenor_four_musicmaker_one],
1681     [(30, 8), (32, 8), tenor_four_musicmaker_one],
1682     [(32, 8), (34, 8), tenor_four_musicmaker_two],
1683     [(34, 8), (36, 8), tenor_four_musicmaker_two],
1684     [(36, 8), (38, 8), tenor_four_musicmaker_two],
1685     [(38, 8), (40, 8), tenor_four_musicmaker_two],
1686     [(40, 8), (42, 8), tenor_four_musicmaker_two],
1687     [(42, 8), (44, 8), tenor_four_musicmaker_two],
1688     [(44, 8), (46, 8), tenor_four_musicmaker_two],
1689     [(46, 8), (48, 8), tenor_four_musicmaker_two],
1690     [(48, 8), (50, 8), tenor_four_musicmaker_one],
1691     [(50, 8), (52, 8), tenor_four_musicmaker_one],
1692     [(52, 8), (54, 8), tenor_four_musicmaker_one],
1693     [(54, 8), (56, 8), tenor_four_musicmaker_one],
1694     [(56, 8), (58, 8), tenor_four_musicmaker_one],
1695     [(58, 8), (60, 8), tenor_four_musicmaker_one],
1696     [(60, 8), (62, 8), tenor_four_musicmaker_one],
1697     [(62, 8), (64, 8), tenor_four_musicmaker_two],
1698     [(64, 8), (66, 8), tenor_four_musicmaker_two],
1699     [(66, 8), (68, 8), tenor_four_musicmaker_two],
1700     [(68, 8), (70, 8), tenor_four_musicmaker_two],
1701     [(70, 8), (72, 8), tenor_four_musicmaker_two],
1702     [(72, 8), (74, 8), tenor_four_musicmaker_two],
1703     [(74, 8), (76, 8), tenor_four_musicmaker_one],
1704     [(76, 8), (78, 8), tenor_four_musicmaker_one],
1705     [(78, 8), (80, 8), tenor_four_musicmaker_one],
1706     [(80, 8), (82, 8), tenor_four_musicmaker_one],
1707     [(82, 8), (84, 8), tenor_four_musicmaker_one],
1708     [(84, 8), (86, 8), tenor_four_musicmaker_two],
1709     [(86, 8), (88, 8), tenor_four_musicmaker_two],
1710     [(88, 8), (90, 8), tenor_four_musicmaker_two],
1711     [(90, 8), (92, 8), tenor_four_musicmaker_two],
1712     [(92, 8), (94, 8), tenor_four_musicmaker_one],
1713     [(94, 8), (96, 8), tenor_four_musicmaker_one],
1714     [(96, 8), (98, 8), tenor_four_musicmaker_one],
1715     [(98, 8), (100, 8), tenor_four_musicmaker_two],
1716     [(100, 8), (102, 8), tenor_four_musicmaker_two],
1717     [(102, 8), (104, 8), tenor_four_musicmaker_one],
1718     [(104, 8), (106, 8), tenor_four_musicmaker_two],
1719     [(106, 8), (108, 8), tenor_four_musicmaker_one],
1720     [(108, 8), (110, 8), tenor_four_musicmaker_one],
1721     [(110, 8), (112, 8), tenor_four_musicmaker_two],
```

```
1722     ]
1723 ])
1724
1725 voice_15_timespan_list = abjad.TimespanList([
1726     abjad.AnnotatedTimespan(
1727         start_offset=start_offset,
1728         stop_offset=stop_offset,
1729         annotation=MusicSpecifier(
1730             music_maker=music_maker,
1731             voice_name='Voice 15',
1732         ),
1733     )
1734     for start_offset, stop_offset, music_maker in [
1735         [(0, 8), (2, 8), tenor_five_musicmaker_two],
1736         [(2, 8), (4, 8), tenor_five_musicmaker_two],
1737         [(4, 8), (6, 8), tenor_five_musicmaker_two],
1738         [(6, 8), (8, 8), tenor_five_musicmaker_one],
1739         [(8, 8), (10, 8), tenor_five_musicmaker_one],
1740         [(10, 8), (12, 8), tenor_five_musicmaker_one],
1741         [(12, 8), (14, 8), tenor_five_musicmaker_two],
1742         [(14, 8), (16, 8), tenor_five_musicmaker_two],
1743         [(16, 8), (18, 8), tenor_five_musicmaker_two],
1744         [(18, 8), (20, 8), tenor_five_musicmaker_two],
1745         [(20, 8), (22, 8), tenor_five_musicmaker_one],
1746         [(22, 8), (24, 8), tenor_five_musicmaker_one],
1747         [(24, 8), (26, 8), tenor_five_musicmaker_one],
1748         [(26, 8), (28, 8), tenor_five_musicmaker_one],
1749         [(28, 8), (30, 8), tenor_five_musicmaker_two],
1750         [(30, 8), (32, 8), tenor_five_musicmaker_two],
1751         [(32, 8), (34, 8), tenor_five_musicmaker_two],
1752         [(34, 8), (36, 8), tenor_five_musicmaker_one],
1753         [(36, 8), (38, 8), tenor_five_musicmaker_one],
1754         [(38, 8), (40, 8), tenor_five_musicmaker_one],
1755         [(40, 8), (42, 8), tenor_five_musicmaker_one],
1756         [(42, 8), (44, 8), tenor_five_musicmaker_two],
1757         [(44, 8), (46, 8), tenor_five_musicmaker_two],
1758         [(46, 8), (48, 8), tenor_five_musicmaker_one],
1759         [(48, 8), (50, 8), tenor_five_musicmaker_one],
1760         [(50, 8), (52, 8), tenor_five_musicmaker_one],
1761         [(52, 8), (54, 8), tenor_five_musicmaker_one],
1762         [(54, 8), (56, 8), tenor_five_musicmaker_one],
1763         [(56, 8), (58, 8), tenor_five_musicmaker_one],
1764         [(58, 8), (60, 8), tenor_five_musicmaker_two],
1765         [(60, 8), (62, 8), tenor_five_musicmaker_one],
1766         [(62, 8), (64, 8), tenor_five_musicmaker_one],
1767         [(64, 8), (66, 8), tenor_five_musicmaker_one],
1768         [(66, 8), (68, 8), tenor_five_musicmaker_one],
1769         [(68, 8), (70, 8), tenor_five_musicmaker_one],
1770         [(70, 8), (72, 8), tenor_five_musicmaker_one],
1771         [(72, 8), (74, 8), tenor_five_musicmaker_one],
1772         [(74, 8), (76, 8), tenor_five_musicmaker_two],
1773         [(76, 8), (78, 8), tenor_five_musicmaker_two],
1774         [(78, 8), (80, 8), tenor_five_musicmaker_two],
1775         [(80, 8), (82, 8), tenor_five_musicmaker_two],
1776         [(82, 8), (84, 8), tenor_five_musicmaker_two],
1777         [(84, 8), (86, 8), tenor_five_musicmaker_two],
1778         [(86, 8), (88, 8), tenor_five_musicmaker_two],
1779         [(88, 8), (90, 8), tenor_five_musicmaker_two],
1780         [(90, 8), (92, 8), tenor_five_musicmaker_one],
```

```

1781     [(92, 8), (94, 8), tenor_five_musicmaker_one],
1782     [(94, 8), (96, 8), tenor_five_musicmaker_one],
1783     [(96, 8), (98, 8), tenor_five_musicmaker_one],
1784     [(98, 8), (100, 8), tenor_five_musicmaker_one],
1785     [(100, 8), (102, 8), tenor_five_musicmaker_one],
1786     [(102, 8), (104, 8), tenor_five_musicmaker_one],
1787     [(104, 8), (106, 8), tenor_five_musicmaker_two],
1788     [(106, 8), (108, 8), tenor_five_musicmaker_two],
1789     [(108, 8), (110, 8), tenor_five_musicmaker_two],
1790     [(110, 8), (112, 8), tenor_five_musicmaker_two],
1791 ]
1792 ])
1793
1794 voice_16_timespan_list = abjad.TimespanList([
1795     abjad.AnnotatedTimespan(
1796         start_offset=start_offset,
1797         stop_offset=stop_offset,
1798         annotation=MusicSpecifier(
1799             music_maker=music_maker,
1800             voice_name='Voice 16',
1801         ),
1802     ),
1803     for start_offset, stop_offset, music_maker in [
1804         [(0, 8), (2, 8), baritone_one_musicmaker_two],
1805         [(2, 8), (4, 8), baritone_one_musicmaker_two],
1806         [(4, 8), (6, 8), baritone_one_musicmaker_two],
1807         [(6, 8), (8, 8), baritone_one_musicmaker_two],
1808         [(8, 8), (10, 8), baritone_one_musicmaker_one],
1809         [(10, 8), (12, 8), baritone_one_musicmaker_one],
1810         [(12, 8), (14, 8), baritone_one_musicmaker_one],
1811         [(14, 8), (16, 8), baritone_one_musicmaker_two],
1812         [(16, 8), (18, 8), baritone_one_musicmaker_two],
1813         [(18, 8), (20, 8), baritone_one_musicmaker_two],
1814         [(20, 8), (22, 8), baritone_one_musicmaker_two],
1815         [(22, 8), (24, 8), baritone_one_musicmaker_one],
1816         [(24, 8), (26, 8), baritone_one_musicmaker_one],
1817         [(26, 8), (28, 8), baritone_one_musicmaker_one],
1818         [(28, 8), (30, 8), baritone_one_musicmaker_two],
1819         [(30, 8), (32, 8), baritone_one_musicmaker_two],
1820         [(32, 8), (34, 8), baritone_one_musicmaker_two],
1821         [(34, 8), (36, 8), baritone_one_musicmaker_two],
1822         [(36, 8), (38, 8), baritone_one_musicmaker_one],
1823         [(38, 8), (40, 8), baritone_one_musicmaker_one],
1824         [(40, 8), (42, 8), baritone_one_musicmaker_one],
1825         [(42, 8), (44, 8), baritone_one_musicmaker_two],
1826         [(44, 8), (46, 8), baritone_one_musicmaker_two],
1827         [(46, 8), (48, 8), baritone_one_musicmaker_two],
1828         [(48, 8), (50, 8), baritone_one_musicmaker_two],
1829         [(50, 8), (52, 8), baritone_one_musicmaker_one],
1830         [(52, 8), (54, 8), baritone_one_musicmaker_one],
1831         [(54, 8), (56, 8), baritone_one_musicmaker_one],
1832         [(56, 8), (58, 8), baritone_one_musicmaker_two],
1833         [(58, 8), (60, 8), baritone_one_musicmaker_two],
1834         [(60, 8), (62, 8), baritone_one_musicmaker_two],
1835         [(62, 8), (64, 8), baritone_one_musicmaker_two],
1836         [(64, 8), (66, 8), baritone_one_musicmaker_one],
1837         [(66, 8), (68, 8), baritone_one_musicmaker_one],
1838         [(68, 8), (70, 8), baritone_one_musicmaker_one],
1839         [(70, 8), (72, 8), baritone_one_musicmaker_two],

```

```

1840     [(72, 8), (74, 8), baritone_one_musicmaker_two],
1841     [(74, 8), (76, 8), baritone_one_musicmaker_two],
1842     [(76, 8), (78, 8), baritone_one_musicmaker_two],
1843     [(78, 8), (80, 8), baritone_one_musicmaker_one],
1844     [(80, 8), (82, 8), baritone_one_musicmaker_one],
1845     [(82, 8), (84, 8), baritone_one_musicmaker_one],
1846     [(84, 8), (86, 8), baritone_one_musicmaker_two],
1847     [(86, 8), (88, 8), baritone_one_musicmaker_two],
1848     [(88, 8), (90, 8), baritone_one_musicmaker_two],
1849     [(90, 8), (92, 8), baritone_one_musicmaker_two],
1850     [(92, 8), (94, 8), baritone_one_musicmaker_one],
1851     [(94, 8), (96, 8), baritone_one_musicmaker_one],
1852     [(96, 8), (98, 8), baritone_one_musicmaker_one],
1853     [(98, 8), (100, 8), baritone_one_musicmaker_two],
1854     [(100, 8), (102, 8), baritone_one_musicmaker_two],
1855     [(102, 8), (104, 8), baritone_one_musicmaker_two],
1856     [(104, 8), (106, 8), baritone_one_musicmaker_two],
1857     [(106, 8), (108, 8), baritone_one_musicmaker_one],
1858     [(108, 8), (110, 8), baritone_one_musicmaker_one],
1859     [(110, 8), (112, 8), baritone_one_musicmaker_one],
1860   ],
1861 ])
1862
1863 voice_17_timespan_list = abjad.TimespanList([
1864     abjad.AnnotatedTimespan(
1865         start_offset=start_offset,
1866         stop_offset=stop_offset,
1867         annotation=MusicSpecifier(
1868             music_maker=music_maker,
1869             voice_name='Voice 17',
1870         ),
1871     ),
1872     for start_offset, stop_offset, music_maker in [
1873         [(0, 8), (2, 8), baritone_two_musicmaker_one],
1874         [(2, 8), (4, 8), baritone_two_musicmaker_one],
1875         [(4, 8), (6, 8), baritone_two_musicmaker_two],
1876         [(6, 8), (8, 8), baritone_two_musicmaker_two],
1877         [(8, 8), (10, 8), baritone_two_musicmaker_two],
1878         [(10, 8), (12, 8), baritone_two_musicmaker_one],
1879         [(12, 8), (14, 8), baritone_two_musicmaker_one],
1880         [(14, 8), (16, 8), baritone_two_musicmaker_one],
1881         [(16, 8), (18, 8), baritone_two_musicmaker_one],
1882         [(18, 8), (20, 8), baritone_two_musicmaker_two],
1883         [(20, 8), (22, 8), baritone_two_musicmaker_two],
1884         [(22, 8), (24, 8), baritone_two_musicmaker_two],
1885         [(24, 8), (26, 8), baritone_two_musicmaker_two],
1886         [(26, 8), (28, 8), baritone_two_musicmaker_two],
1887         [(28, 8), (30, 8), baritone_two_musicmaker_one],
1888         [(30, 8), (32, 8), baritone_two_musicmaker_one],
1889         [(32, 8), (34, 8), baritone_two_musicmaker_one],
1890         [(34, 8), (36, 8), baritone_two_musicmaker_one],
1891         [(36, 8), (38, 8), baritone_two_musicmaker_one],
1892         [(38, 8), (40, 8), baritone_two_musicmaker_one],
1893         [(40, 8), (42, 8), baritone_two_musicmaker_two],
1894         [(42, 8), (44, 8), baritone_two_musicmaker_two],
1895         [(44, 8), (46, 8), baritone_two_musicmaker_two],
1896         [(46, 8), (48, 8), baritone_two_musicmaker_two],
1897         [(48, 8), (50, 8), baritone_two_musicmaker_two],
1898         [(50, 8), (52, 8), baritone_two_musicmaker_two],

```

```

1899     [(52, 8), (54, 8), baritone_two_musicmaker_two],
1900     [(54, 8), (56, 8), baritone_two_musicmaker_one],
1901     [(56, 8), (58, 8), baritone_two_musicmaker_one],
1902     [(58, 8), (60, 8), baritone_two_musicmaker_one],
1903     [(60, 8), (62, 8), baritone_two_musicmaker_one],
1904     [(62, 8), (64, 8), baritone_two_musicmaker_one],
1905     [(64, 8), (66, 8), baritone_two_musicmaker_one],
1906     [(66, 8), (68, 8), baritone_two_musicmaker_one],
1907     [(68, 8), (70, 8), baritone_two_musicmaker_two],
1908     [(70, 8), (72, 8), baritone_two_musicmaker_two],
1909     [(72, 8), (74, 8), baritone_two_musicmaker_two],
1910     [(74, 8), (76, 8), baritone_two_musicmaker_two],
1911     [(76, 8), (78, 8), baritone_two_musicmaker_two],
1912     [(78, 8), (80, 8), baritone_two_musicmaker_two],
1913     [(80, 8), (82, 8), baritone_two_musicmaker_two],
1914     [(82, 8), (84, 8), baritone_two_musicmaker_one],
1915     [(84, 8), (86, 8), baritone_two_musicmaker_one],
1916     [(86, 8), (88, 8), baritone_two_musicmaker_one],
1917     [(88, 8), (90, 8), baritone_two_musicmaker_one],
1918     [(90, 8), (92, 8), baritone_two_musicmaker_one],
1919     [(92, 8), (94, 8), baritone_two_musicmaker_one],
1920     [(94, 8), (96, 8), baritone_two_musicmaker_one],
1921     [(96, 8), (98, 8), baritone_two_musicmaker_two],
1922     [(98, 8), (100, 8), baritone_two_musicmaker_two],
1923     [(100, 8), (102, 8), baritone_two_musicmaker_two],
1924     [(102, 8), (104, 8), baritone_two_musicmaker_two],
1925     [(104, 8), (106, 8), baritone_two_musicmaker_two],
1926     [(106, 8), (108, 8), baritone_two_musicmaker_one],
1927     [(108, 8), (110, 8), baritone_two_musicmaker_one],
1928     [(110, 8), (112, 8), baritone_two_musicmaker_one],
1929 ]
1930 ])
1931
1932 voice_18_timespan_list = abjad.TimespanList([
1933     abjad.AnnotatedTimespan(
1934         start_offset=start_offset,
1935         stop_offset=stop_offset,
1936         annotation=MusicSpecifier(
1937             music_maker=music_maker,
1938             voice_name='Voice 18',
1939         ),
1940     )
1941     for start_offset, stop_offset, music_maker in [
1942         [(0, 8), (2, 8), baritone_three_musicmaker_one],
1943         [(2, 8), (4, 8), baritone_three_musicmaker_one],
1944         [(4, 8), (6, 8), baritone_three_musicmaker_one],
1945         [(6, 8), (8, 8), baritone_three_musicmaker_one],
1946         [(8, 8), (10, 8), baritone_three_musicmaker_two],
1947         [(10, 8), (12, 8), baritone_three_musicmaker_two],
1948         [(12, 8), (14, 8), baritone_three_musicmaker_two],
1949         [(14, 8), (16, 8), baritone_three_musicmaker_two],
1950         [(16, 8), (18, 8), baritone_three_musicmaker_one],
1951         [(18, 8), (20, 8), baritone_three_musicmaker_one],
1952         [(20, 8), (22, 8), baritone_three_musicmaker_one],
1953         [(22, 8), (24, 8), baritone_three_musicmaker_one],
1954         [(24, 8), (26, 8), baritone_three_musicmaker_two],
1955         [(26, 8), (28, 8), baritone_three_musicmaker_two],
1956         [(28, 8), (30, 8), baritone_three_musicmaker_two],
1957         [(30, 8), (32, 8), baritone_three_musicmaker_two],

```

```

1958     [(32, 8), (34, 8), baritone_three_musicmaker_one],
1959     [(34, 8), (36, 8), baritone_three_musicmaker_one],
1960     [(36, 8), (38, 8), baritone_three_musicmaker_one],
1961     [(38, 8), (40, 8), baritone_three_musicmaker_one],
1962     [(40, 8), (42, 8), baritone_three_musicmaker_two],
1963     [(42, 8), (44, 8), baritone_three_musicmaker_two],
1964     [(44, 8), (46, 8), baritone_three_musicmaker_two],
1965     [(46, 8), (48, 8), baritone_three_musicmaker_two],
1966     [(48, 8), (50, 8), baritone_three_musicmaker_one],
1967     [(50, 8), (52, 8), baritone_three_musicmaker_one],
1968     [(52, 8), (54, 8), baritone_three_musicmaker_one],
1969     [(54, 8), (56, 8), baritone_three_musicmaker_one],
1970     [(56, 8), (58, 8), baritone_three_musicmaker_two],
1971     [(58, 8), (60, 8), baritone_three_musicmaker_two],
1972     [(60, 8), (62, 8), baritone_three_musicmaker_two],
1973     [(62, 8), (64, 8), baritone_three_musicmaker_two],
1974     [(64, 8), (66, 8), baritone_three_musicmaker_one],
1975     [(66, 8), (68, 8), baritone_three_musicmaker_one],
1976     [(68, 8), (70, 8), baritone_three_musicmaker_one],
1977     [(70, 8), (72, 8), baritone_three_musicmaker_one],
1978     [(72, 8), (74, 8), baritone_three_musicmaker_two],
1979     [(74, 8), (76, 8), baritone_three_musicmaker_two],
1980     [(76, 8), (78, 8), baritone_three_musicmaker_two],
1981     [(78, 8), (80, 8), baritone_three_musicmaker_two],
1982     [(80, 8), (82, 8), baritone_three_musicmaker_one],
1983     [(82, 8), (84, 8), baritone_three_musicmaker_one],
1984     [(84, 8), (86, 8), baritone_three_musicmaker_one],
1985     [(86, 8), (88, 8), baritone_three_musicmaker_one],
1986     [(88, 8), (90, 8), baritone_three_musicmaker_two],
1987     [(90, 8), (92, 8), baritone_three_musicmaker_two],
1988     [(92, 8), (94, 8), baritone_three_musicmaker_two],
1989     [(94, 8), (96, 8), baritone_three_musicmaker_two],
1990     [(96, 8), (98, 8), baritone_three_musicmaker_one],
1991     [(98, 8), (100, 8), baritone_three_musicmaker_one],
1992     [(100, 8), (102, 8), baritone_three_musicmaker_one],
1993     [(102, 8), (104, 8), baritone_three_musicmaker_one],
1994     [(104, 8), (106, 8), baritone_three_musicmaker_two],
1995     [(106, 8), (108, 8), baritone_three_musicmaker_two],
1996     [(108, 8), (110, 8), baritone_three_musicmaker_two],
1997     [(110, 8), (112, 8), baritone_three_musicmaker_two],
1998 ]
1999 ])
2000
2001 voice_19_timespan_list = abjad.TimespanList([
2002     abjad.AnnotatedTimespan(
2003         start_offset=start_offset,
2004         stop_offset=stop_offset,
2005         annotation=MusicSpecifier(
2006             music_maker=music_maker,
2007             voice_name='Voice 19',
2008         ),
2009     )
2010     for start_offset, stop_offset, music_maker in [
2011         [(0, 8), (2, 8), bass_one_musicmaker_two],
2012         [(2, 8), (4, 8), bass_one_musicmaker_two],
2013         [(4, 8), (6, 8), bass_one_musicmaker_two],
2014         [(6, 8), (8, 8), bass_one_musicmaker_one],
2015         [(8, 8), (10, 8), bass_one_musicmaker_one],
2016         [(10, 8), (12, 8), bass_one_musicmaker_two],

```

```

2017     [(12, 8), (14, 8), bass_one_musicmaker_two],
2018     [(14, 8), (16, 8), bass_one_musicmaker_two],
2019     [(16, 8), (18, 8), bass_one_musicmaker_two],
2020     [(18, 8), (20, 8), bass_one_musicmaker_one],
2021     [(20, 8), (22, 8), bass_one_musicmaker_two],
2022     [(22, 8), (24, 8), bass_one_musicmaker_two],
2023     [(24, 8), (26, 8), bass_one_musicmaker_two],
2024     [(26, 8), (28, 8), bass_one_musicmaker_two],
2025     [(28, 8), (30, 8), bass_one_musicmaker_two],
2026     [(30, 8), (32, 8), bass_one_musicmaker_one],
2027     [(32, 8), (34, 8), bass_one_musicmaker_one],
2028     [(34, 8), (36, 8), bass_one_musicmaker_one],
2029     [(36, 8), (38, 8), bass_one_musicmaker_one],
2030     [(38, 8), (40, 8), bass_one_musicmaker_one],
2031     [(40, 8), (42, 8), bass_one_musicmaker_one],
2032     [(42, 8), (44, 8), bass_one_musicmaker_one],
2033     [(44, 8), (46, 8), bass_one_musicmaker_one],
2034     [(46, 8), (48, 8), bass_one_musicmaker_two],
2035     [(48, 8), (50, 8), bass_one_musicmaker_two],
2036     [(50, 8), (52, 8), bass_one_musicmaker_two],
2037     [(52, 8), (54, 8), bass_one_musicmaker_two],
2038     [(54, 8), (56, 8), bass_one_musicmaker_two],
2039     [(56, 8), (58, 8), bass_one_musicmaker_two],
2040     [(58, 8), (60, 8), bass_one_musicmaker_one],
2041     [(60, 8), (62, 8), bass_one_musicmaker_one],
2042     [(62, 8), (64, 8), bass_one_musicmaker_one],
2043     [(64, 8), (68, 8), bass_one_musicmaker_one],
2044     [(68, 8), (70, 8), bass_one_musicmaker_one],
2045     [(70, 8), (72, 8), bass_one_musicmaker_one],
2046     [(72, 8), (74, 8), bass_one_musicmaker_one],
2047     [(74, 8), (76, 8), bass_one_musicmaker_two],
2048     [(76, 8), (78, 8), bass_one_musicmaker_two],
2049     [(78, 8), (80, 8), bass_one_musicmaker_two],
2050     [(80, 8), (82, 8), bass_one_musicmaker_two],
2051     [(82, 8), (84, 8), bass_one_musicmaker_two],
2052     [(84, 8), (86, 8), bass_one_musicmaker_two],
2053     [(86, 8), (88, 8), bass_one_musicmaker_two],
2054     [(88, 8), (90, 8), bass_one_musicmaker_one],
2055     [(90, 8), (92, 8), bass_one_musicmaker_one],
2056     [(92, 8), (94, 8), bass_one_musicmaker_one],
2057     [(94, 8), (96, 8), bass_one_musicmaker_one],
2058     [(96, 8), (98, 8), bass_one_musicmaker_one],
2059     [(98, 8), (100, 8), bass_one_musicmaker_one],
2060     [(100, 8), (102, 8), bass_one_musicmaker_two],
2061     [(102, 8), (104, 8), bass_one_musicmaker_two],
2062     [(104, 8), (106, 8), bass_one_musicmaker_two],
2063     [(106, 8), (108, 8), bass_one_musicmaker_two],
2064     [(108, 8), (110, 8), bass_one_musicmaker_two],
2065     [(110, 8), (112, 8), bass_one_musicmaker_two],
2066 ]
2067 ])
2068
2069 voice_20_timespan_list = abjad.TimespanList([
2070     abjad.AnnotatedTimespan(
2071         start_offset=start_offset,
2072         stop_offset=stop_offset,
2073         annotation=MusicSpecifier(
2074             music_maker=music_maker,
2075             voice_name='Voice 20',

```

```

2076     ),
2077
2078     for start_offset, stop_offset, music_maker in [
2079         [(0, 8), (2, 8), bass_two_musicmaker_two],
2080         [(2, 8), (4, 8), bass_two_musicmaker_two],
2081         [(4, 8), (6, 8), bass_two_musicmaker_two],
2082         [(6, 8), (8, 8), bass_two_musicmaker_one],
2083         [(8, 8), (10, 8), bass_two_musicmaker_one],
2084         [(10, 8), (12, 8), bass_two_musicmaker_two],
2085         [(12, 8), (14, 8), bass_two_musicmaker_two],
2086         [(14, 8), (16, 8), bass_two_musicmaker_two],
2087         [(16, 8), (18, 8), bass_two_musicmaker_two],
2088         [(18, 8), (20, 8), bass_two_musicmaker_one],
2089         [(20, 8), (22, 8), bass_two_musicmaker_two],
2090         [(22, 8), (24, 8), bass_two_musicmaker_two],
2091         [(24, 8), (26, 8), bass_two_musicmaker_two],
2092         [(26, 8), (28, 8), bass_two_musicmaker_two],
2093         [(28, 8), (30, 8), bass_two_musicmaker_two],
2094         [(30, 8), (32, 8), bass_two_musicmaker_one],
2095         [(32, 8), (34, 8), bass_two_musicmaker_one],
2096         [(34, 8), (36, 8), bass_two_musicmaker_one],
2097         [(36, 8), (38, 8), bass_two_musicmaker_one],
2098         [(38, 8), (40, 8), bass_two_musicmaker_one],
2099         [(40, 8), (42, 8), bass_two_musicmaker_one],
2100         [(42, 8), (44, 8), bass_two_musicmaker_one],
2101         [(44, 8), (46, 8), bass_two_musicmaker_one],
2102         [(46, 8), (48, 8), bass_two_musicmaker_two],
2103         [(48, 8), (50, 8), bass_two_musicmaker_two],
2104         [(50, 8), (52, 8), bass_two_musicmaker_two],
2105         [(52, 8), (54, 8), bass_two_musicmaker_two],
2106         [(54, 8), (56, 8), bass_two_musicmaker_two],
2107         [(56, 8), (58, 8), bass_two_musicmaker_two],
2108         [(58, 8), (60, 8), bass_two_musicmaker_one],
2109         [(60, 8), (62, 8), bass_two_musicmaker_one],
2110         [(62, 8), (64, 8), bass_two_musicmaker_one],
2111         [(64, 8), (68, 8), bass_two_musicmaker_one],
2112         [(68, 8), (70, 8), bass_two_musicmaker_one],
2113         [(70, 8), (72, 8), bass_two_musicmaker_one],
2114         [(72, 8), (74, 8), bass_two_musicmaker_one],
2115         [(74, 8), (76, 8), bass_two_musicmaker_two],
2116         [(76, 8), (78, 8), bass_two_musicmaker_two],
2117         [(78, 8), (80, 8), bass_two_musicmaker_two],
2118         [(80, 8), (82, 8), bass_two_musicmaker_two],
2119         [(82, 8), (84, 8), bass_two_musicmaker_two],
2120         [(84, 8), (86, 8), bass_two_musicmaker_two],
2121         [(86, 8), (88, 8), bass_two_musicmaker_two],
2122         [(88, 8), (90, 8), bass_two_musicmaker_one],
2123         [(90, 8), (92, 8), bass_two_musicmaker_one],
2124         [(92, 8), (94, 8), bass_two_musicmaker_one],
2125         [(94, 8), (96, 8), bass_two_musicmaker_one],
2126         [(96, 8), (98, 8), bass_two_musicmaker_one],
2127         [(98, 8), (100, 8), bass_two_musicmaker_one],
2128         [(100, 8), (102, 8), bass_two_musicmaker_two],
2129         [(102, 8), (104, 8), bass_two_musicmaker_two],
2130         [(104, 8), (106, 8), bass_two_musicmaker_two],
2131         [(106, 8), (108, 8), bass_two_musicmaker_two],
2132         [(108, 8), (110, 8), bass_two_musicmaker_two],
2133         [(110, 8), (112, 8), bass_two_musicmaker_two],
2134     ]

```

```
2135 ])
2136
2137 voice_21_timespan_list = abjad.TimespanList([
2138     abjad.AnnotatedTimespan(
2139         start_offset=start_offset,
2140         stop_offset=stop_offset,
2141         annotation=MusicSpecifier(
2142             music_maker=music_maker,
2143             voice_name='Voice 21',
2144         ),
2145     )
2146     for start_offset, stop_offset, music_maker in [
2147         [(0, 8), (2, 8), contrabass_musicmaker_two],
2148         [(2, 8), (4, 8), contrabass_musicmaker_two],
2149         [(4, 8), (6, 8), contrabass_musicmaker_two],
2150         [(6, 8), (8, 8), contrabass_musicmaker_one],
2151         [(8, 8), (10, 8), contrabass_musicmaker_one],
2152         [(10, 8), (12, 8), contrabass_musicmaker_two],
2153         [(12, 8), (14, 8), contrabass_musicmaker_two],
2154         [(14, 8), (16, 8), contrabass_musicmaker_two],
2155         [(16, 8), (18, 8), contrabass_musicmaker_two],
2156         [(18, 8), (20, 8), contrabass_musicmaker_one],
2157         [(20, 8), (22, 8), contrabass_musicmaker_two],
2158         [(22, 8), (24, 8), contrabass_musicmaker_two],
2159         [(24, 8), (26, 8), contrabass_musicmaker_two],
2160         [(26, 8), (28, 8), contrabass_musicmaker_two],
2161         [(28, 8), (30, 8), contrabass_musicmaker_two],
2162         [(30, 8), (32, 8), contrabass_musicmaker_one],
2163         [(32, 8), (34, 8), contrabass_musicmaker_one],
2164         [(34, 8), (36, 8), contrabass_musicmaker_one],
2165         [(36, 8), (38, 8), contrabass_musicmaker_one],
2166         [(38, 8), (40, 8), contrabass_musicmaker_one],
2167         [(40, 8), (42, 8), contrabass_musicmaker_one],
2168         [(42, 8), (44, 8), contrabass_musicmaker_one],
2169         [(44, 8), (46, 8), contrabass_musicmaker_one],
2170         [(46, 8), (48, 8), contrabass_musicmaker_two],
2171         [(48, 8), (50, 8), contrabass_musicmaker_two],
2172         [(50, 8), (52, 8), contrabass_musicmaker_two],
2173         [(52, 8), (54, 8), contrabass_musicmaker_two],
2174         [(54, 8), (56, 8), contrabass_musicmaker_two],
2175         [(56, 8), (58, 8), contrabass_musicmaker_two],
2176         [(58, 8), (60, 8), contrabass_musicmaker_one],
2177         [(60, 8), (62, 8), contrabass_musicmaker_one],
2178         [(62, 8), (64, 8), contrabass_musicmaker_one],
2179         [(64, 8), (68, 8), contrabass_musicmaker_one],
2180         [(68, 8), (70, 8), contrabass_musicmaker_one],
2181         [(70, 8), (72, 8), contrabass_musicmaker_one],
2182         [(72, 8), (74, 8), contrabass_musicmaker_one],
2183         [(74, 8), (76, 8), contrabass_musicmaker_two],
2184         [(76, 8), (78, 8), contrabass_musicmaker_two],
2185         [(78, 8), (80, 8), contrabass_musicmaker_two],
2186         [(80, 8), (82, 8), contrabass_musicmaker_two],
2187         [(82, 8), (84, 8), contrabass_musicmaker_two],
2188         [(84, 8), (86, 8), contrabass_musicmaker_two],
2189         [(86, 8), (88, 8), contrabass_musicmaker_two],
2190         [(88, 8), (90, 8), contrabass_musicmaker_one],
2191         [(90, 8), (92, 8), contrabass_musicmaker_one],
2192         [(92, 8), (94, 8), contrabass_musicmaker_one],
2193         [(94, 8), (96, 8), contrabass_musicmaker_one],
```

```

2194     [(96, 8), (98, 8), contrabass_musicmaker_one],
2195     [(98, 8), (100, 8), contrabass_musicmaker_one],
2196     [(100, 8), (102, 8), contrabass_musicmaker_two],
2197     [(102, 8), (104, 8), contrabass_musicmaker_two],
2198     [(104, 8), (106, 8), contrabass_musicmaker_two],
2199     [(106, 8), (108, 8), contrabass_musicmaker_two],
2200     [(108, 8), (110, 8), contrabass_musicmaker_two],
2201     [(110, 8), (112, 8), contrabass_musicmaker_two],
2202 ]
2203 ])
2204
2205 all_timespan_lists = {
2206     'Voice 1': voice_1_timespan_list,
2207     'Voice 2': voice_2_timespan_list,
2208     'Voice 3': voice_3_timespan_list,
2209     'Voice 4': voice_4_timespan_list,
2210     'Voice 5': voice_5_timespan_list,
2211     'Voice 6': voice_6_timespan_list,
2212     'Voice 7': voice_7_timespan_list,
2213     'Voice 8': voice_8_timespan_list,
2214     'Voice 9': voice_9_timespan_list,
2215     'Voice 10': voice_10_timespan_list,
2216     'Voice 11': voice_11_timespan_list,
2217     'Voice 12': voice_12_timespan_list,
2218     'Voice 13': voice_13_timespan_list,
2219     'Voice 14': voice_14_timespan_list,
2220     'Voice 15': voice_15_timespan_list,
2221     'Voice 16': voice_16_timespan_list,
2222     'Voice 17': voice_17_timespan_list,
2223     'Voice 18': voice_18_timespan_list,
2224     'Voice 19': voice_19_timespan_list,
2225     'Voice 20': voice_20_timespan_list,
2226     'Voice 21': voice_21_timespan_list,
2227 }
2228
2229 global_timespan = abjad.Timespan(
2230     start_offset=0,
2231     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
2232 )
2233
2234 for voice_name, timespan_list in all_timespan_lists.items():
2235     silences = abjad.TimespanList([global_timespan])
2236     silences.extend(timespan_list)
2237     silences.sort()
2238     silences.compute_logical_xor()
2239     for silence_timespan in silences:
2240         timespan_list.append(
2241             abjad.AnnotatedTimespan(
2242                 start_offset=silence_timespan.start_offset,
2243                 stop_offset=silence_timespan.stop_offset,
2244                 annotation=MusicSpecifier(
2245                     music_maker=None,
2246                     voice_name=voice_name,
2247                 ),
2248             )
2249         )
2250     timespan_list.sort()
2251
2252 for voice_name, timespan_list in all_timespan_lists.items():

```

```

2253 shards = timespan_list.split_at_offsets(bounds)
2254 split_timespan_list = abjad.TimespanList()
2255 for shard in shards:
2256     split_timespan_list.extend(shard)
2257 split_timespan_list.sort()
2258 all_timespan_lists[voice_name] = timespan_list
2259
2260 score = abjad.Score([
2261     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
2262     abjad.StaffGroup(
2263         [
2264             abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
2265                         lilypond_type='Staff',),
2266             abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
2267                         lilypond_type='Staff',),
2268             abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
2269                         lilypond_type='Staff',),
2270             abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
2271                         lilypond_type='Staff',),
2272             abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
2273                         lilypond_type='Staff',),
2274             abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
2275                         lilypond_type='Staff',),
2276             abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
2277                         lilypond_type='Staff',),
2278             abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
2279                         lilypond_type='Staff',),
2280             abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
2281                         lilypond_type='Staff',),
2282             abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
2283                         lilypond_type='Staff',),
2284             abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
2285                         lilypond_type='Staff',),
2286             abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
2287                         lilypond_type='Staff',),
2288             abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
2289                         lilypond_type='Staff',),
2290             abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
2291                         lilypond_type='Staff',),
2292             abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
2293                         lilypond_type='Staff',),
2294             abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
2295                         lilypond_type='Staff',),
2296             abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
2297                         lilypond_type='Staff',),
2298             abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
2299                         lilypond_type='Staff',),
2300             abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
2301                         lilypond_type='Staff',),
2302             abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
2303                         lilypond_type='Staff',),
2304             abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
2305                         lilypond_type='Staff',),
2306         ],
2307         name='Staff Group',
2308     )
2309 ])
2310 for time_signature in time_signatures:

```

```

2291     skip = abjad.Skip(1, multiplier=(time_signature))
2292     abjad.attach(time_signature, skip)
2293     score['Global Context'].append(skip)
2294
2295 print('Making containers ...')
2296
2297 def make_container(music_maker, durations):
2298     selections = music_maker(durations)
2299     container = abjad.Container([])
2300     container.extend(selections)
2301     return container
2302
2303 def key_function(timespan):
2304     return timespan.annotation.music_maker or silence_maker
2305
2306 for voice_name, timespan_list in all_timespan_lists.items():
2307     for music_maker, grouper in itertools.groupby(
2308         timespan_list,
2309         key=key_function,
2310     ):
2311         durations = [timespan.duration for timespan in grouper]
2312         container = make_container(music_maker, durations)
2313         voice = score[voice_name]
2314         voice.append(container)
2315
2316 print('Splitting and rewriting ...')
2317 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
2318     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
2319         time_signature = time_signatures[i]
2320         abjad.mutate(shard).rewrite_meter(time_signature)
2321
2322 print('Beaming runs ...')
2323 for voice in abjad.select(score).components(abjad.Voice):
2324     for run in abjad.select(voice).runs():
2325         specifier = abjadext.rmakers.BeamSpecifier(
2326             beam_each_division=False,
2327         )
2328         specifier(run)
2329     abjad.beam(voice[:, beam_lone_notes=False, beam_rests=False])
2330
2331 print('Stopping Hairpins ...')
2332 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2333     for rest in abjad.iterate(staff).components(abjad.Rest):
2334         previous_leaf = abjad.inspect(rest).leaf(-1)
2335         if isinstance(previous_leaf, abjad.Note):
2336             abjad.attach(abjad.StopHairpin(), rest)
2337         elif isinstance(previous_leaf, abjad.Chord):
2338             abjad.attach(abjad.StopHairpin(), rest)
2339         elif isinstance(previous_leaf, abjad.Rest):
2340             pass
2341
2342 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2343     first_leaf = abjad.select(staff).leaves()[0]
2344     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
2345     abjad.attach(stop, first_leaf)
2346
2347 print('Adding attachments ...')
2348 bar_line = abjad.BarLine('||')
2349 metro = abjad.MetronomeMark((1, 4), 90)

```

```

2350 markup = abjad.Markup(r'\bold { I }')
2351 mark = abjad.RehearsalMark(markup=markup)
2352
2353 instruments = cyc([
2354     abjad.SopranoSaxophone(),
2355     abjad.SopranoSaxophone(),
2356     abjad.SopranoSaxophone(),
2357     abjad.SopranoSaxophone(),
2358     abjad.AltoSaxophone(),
2359     abjad.AltoSaxophone(),
2360     abjad.AltoSaxophone(),
2361     abjad.AltoSaxophone(),
2362     abjad.AltoSaxophone(),
2363     abjad.AltoSaxophone(),
2364     abjad.TenorSaxophone(),
2365     abjad.TenorSaxophone(),
2366     abjad.TenorSaxophone(),
2367     abjad.TenorSaxophone(),
2368     abjad.TenorSaxophone(),
2369     abjad.BaritoneSaxophone(),
2370     abjad.BaritoneSaxophone(),
2371     abjad.BaritoneSaxophone(),
2372     abjad.BassSaxophone(),
2373     abjad.BassSaxophone(),
2374     abjad.ContrabassSaxophone(),
2375 ])
2376
2377 abbreviations = cyc([
2378     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
2379     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
2380     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
2381     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
2382     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
2383     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
2384     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
2385     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
2386     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
2387     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
2388     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
2389     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
2390     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
2391     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
2392     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
2393     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
2394     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
2395     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
2396     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
2397     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
2398     abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
2399 ])
2400
2401 names = cyc([
2402     abjad.StartMarkup(markup=abjad.Markup('Soprano'),),
2403     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
2404     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
2405     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
2406     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
2407     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
2408     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),

```

```

2409 abjad.StartMarkup(markup=abjad.Markup('Alto 4')),,
2410 abjad.StartMarkup(markup=abjad.Markup('Alto 5')),,
2411 abjad.StartMarkup(markup=abjad.Markup('Alto 6')),,
2412 abjad.StartMarkup(markup=abjad.Markup('Tenor 1')),,
2413 abjad.StartMarkup(markup=abjad.Markup('Tenor 2')),,
2414 abjad.StartMarkup(markup=abjad.Markup('Tenor 3')),,
2415 abjad.StartMarkup(markup=abjad.Markup('Tenor 4')),,
2416 abjad.StartMarkup(markup=abjad.Markup('Tenor 5')),,
2417 abjad.StartMarkup(markup=abjad.Markup('Baritone 1')),,
2418 abjad.StartMarkup(markup=abjad.Markup('Baritone 2')),,
2419 abjad.StartMarkup(markup=abjad.Markup('Baritone 3')),,
2420 abjad.StartMarkup(markup=abjad.Markup('Bass 1')),,
2421 abjad.StartMarkup(markup=abjad.Markup('Bass 2')),,
2422 abjad.StartMarkup(markup=abjad.Markup('Contrabass')),,
2423 ])
2424
2425 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
2426     leaf1 = abjad.select(staff).leaves()[0]
2427     abjad.attach(next(instruments), leaf1)
2428     abjad.attach(next(abbreviations), leaf1)
2429     abjad.attach(next(names), leaf1)
2430
2431 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
2432     leaf1 = abjad.select(staff).leaves()[0]
2433     last_leaf = abjad.select(staff).leaves()[-1]
2434     abjad.attach(metro, leaf1)
2435     abjad.attach(bar_line, last_leaf)
2436
2437 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
2438     leaf1 = abjad.select(staff).leaves()[0]
2439     abjad.attach(mark, leaf1)
2440
2441 score_file = abjad.LilyPondFile.new(
2442     score,
2443     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2444     _stylesheets/abjad.ily'],
2445 )
2446
2447 abjad.SegmentMaker.comment_measure_numbers(score)
2448 ######
2449 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_I'
2450 pdf_path = f'{directory}/Section_I.pdf'
2451 path = pathlib.Path('Section_I.pdf')
2452 if path.exists():
2453     print(f'Removing {pdf_path} ...')
2454     path.unlink()
2455 time_1 = time.time()
2456 print(f'Persisting {pdf_path} ...')
2457 result = abjad.persist(score_file).as_pdf(pdf_path)
2458 print(result[0])
2459 print(result[1])
2460 print(result[2])
2461 success = result[3]
2462 if success is False:
2463     print('LilyPond failed!')
2464 time_2 = time.time()
2465 total_time = time_2 - time_1
2466 print(f'Total time: {total_time} seconds')

```

```

2467 if path.exists():
2468     print(f'Opening {pdf_path} ...')
2469     os.system(f'open {pdf_path}')
2470 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_I/Section_I.ly').readlines()
2471 open('/Users/evansdsg2/Scores/guerrero/Build/Section_I.ly', 'w').writelines(
2472     score_lines[15:-1])
2473
2474 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
2475     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2476     signature_copy = abjad.mutate(signatures).copy()
2477     staff_copy = abjad.mutate(staff).copy()
2478     part = abjad.Score()
2479     part.insert(0, staff)
2480     part.insert(0, signature_copy)
2481     part_file = abjad.LilyPondFile.new(
2482         part,
2483         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/_stylesheets/abjad.ily'],
2484         )
2485     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano1'
2486     pdf_path = f'{directory}/Section_I.pdf'
2487     path = pathlib.Path('Section_I.pdf')
2488     if path.exists():
2489         print(f'Removing {pdf_path} ...')
2490         path.unlink()
2491     time_1 = time.time()
2492     print(f'Persisting {pdf_path} ...')
2493     result = abjad.persist(part_file).as_pdf(pdf_path)
2494     print(result[0])
2495     print(result[1])
2496     print(result[2])
2497     success = result[3]
2498     if success is False:
2499         print('LilyPond failed!')
2500     time_2 = time.time()
2501     total_time = time_2 - time_1
2502     print(f'Total time: {total_time} seconds')
2503     if path.exists():
2504         print(f'Opening {pdf_path} ...')
2505         os.system(f'open {pdf_path}')
2506     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano1/Section_I.ly').readlines()
2507     open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.')soprano1/Section_I.ly', 'w').writelines(part_lines[15:-1])
2508
2509 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
2510     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2511     signature_copy = abjad.mutate(signatures).copy()
2512     staff_copy = abjad.mutate(staff).copy()
2513     part = abjad.Score()
2514     part.insert(0, staff)
2515     part.insert(0, signature_copy)
2516     part_file = abjad.LilyPondFile.new(
2517         part,
2518         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/_stylesheets/abjad.ily'],
2519         )
2520     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.')soprano1'

```

```

2520 pdf_path = f'{directory}/Section_I.pdf'
2521 path = pathlib.Path('Section_I.pdf')
2522 if path.exists():
2523     print(f'Removing {pdf_path} ...')
2524     path.unlink()
2525 time_1 = time.time()
2526 print(f'Persisting {pdf_path} ...')
2527 result = abjad.persist(part_file).as_pdf(pdf_path)
2528 print(result[0])
2529 print(result[1])
2530 print(result[2])
2531 success = result[3]
2532 if success is False:
2533     print('LilyPond failed!')
2534 time_2 = time.time()
2535 total_time = time_2 - time_1
2536 print(f'Total time: {total_time} seconds')
2537 if path.exists():
2538     print(f'Opening {pdf_path} ...')
2539     os.system(f'open {pdf_path}')
2540 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/
Section_I.ly').readlines()
2541 open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1/Section_I.ly',
'w').writelines(part_lines[15:-1])
2542
2543 for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
2544     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2545     signature_copy = abjad.mutate(signatures).copy()
2546     staff_copy = abjad.mutate(staff).copy()
2547     part = abjad.Score()
2548     part.insert(0, staff)
2549     part.insert(0, signature_copy)
2550     part_file = abjad.LilyPondFile.new(
2551         part,
2552         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2553         )
2554 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2'
2555 pdf_path = f'{directory}/Section_I.pdf'
2556 path = pathlib.Path('Section_I.pdf')
2557 if path.exists():
2558     print(f'Removing {pdf_path} ...')
2559     path.unlink()
2560 time_1 = time.time()
2561 print(f'Persisting {pdf_path} ...')
2562 result = abjad.persist(part_file).as_pdf(pdf_path)
2563 print(result[0])
2564 print(result[1])
2565 print(result[2])
2566 success = result[3]
2567 if success is False:
2568     print('LilyPond failed!')
2569 time_2 = time.time()
2570 total_time = time_2 - time_1
2571 print(f'Total time: {total_time} seconds')
2572 if path.exists():
2573     print(f'Opening {pdf_path} ...')
2574     os.system(f'open {pdf_path}')
2575 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.) soprano2/

```

```

2576     Section_I.ly').readlines()
2577     open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/Section_I.ly',
2578         'w').writelines(part_lines[15:-1])
2579
2580 for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
2581     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2582     signature_copy = abjad.mutate(signatures).copy()
2583     staff_copy = abjad.mutate(staff).copy()
2584     part = abjad.Score()
2585     part.insert(0, staff)
2586     part.insert(0, signature_copy)
2587     part_file = abjad.LilyPondFile.new(
2588         part,
2589         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2590         _stylesheets/abjad.ily'],
2591         )
2592     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3'
2593     pdf_path = f'{directory}/Section_I.pdf'
2594     path = pathlib.Path('Section_I.pdf')
2595     if path.exists():
2596         print(f'Removing {pdf_path} ...')
2597         path.unlink()
2598     time_1 = time.time()
2599     print(f'Persisting {pdf_path} ...')
2600     result = abjad.persist(part_file).as_pdf(pdf_path)
2601     print(result[0])
2602     print(result[1])
2603     print(result[2])
2604     success = result[3]
2605     if success is False:
2606         print('LilyPond failed!')
2607     time_2 = time.time()
2608     total_time = time_2 - time_1
2609     print(f'Total time: {total_time} seconds')
2610     if path.exists():
2611         print(f'Opening {pdf_path} ...')
2612         os.system(f'open {pdf_path}')
2613     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/
2614     Section_I.ly').readlines()
2615     open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/Section_I.ly',
2616         'w').writelines(part_lines[15:-1])
2617
2618 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
2619     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2620     signature_copy = abjad.mutate(signatures).copy()
2621     staff_copy = abjad.mutate(staff).copy()
2622     part = abjad.Score()
2623     part.insert(0, staff)
2624     part.insert(0, signature_copy)
2625     part_file = abjad.LilyPondFile.new(
2626         part,
2627         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2628         _stylesheets/abjad.ily'],
2629         )
2630     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1'
2631     pdf_path = f'{directory}/Section_I.pdf'
2632     path = pathlib.Path('Section_I.pdf')
2633     if path.exists():
2634         print(f'Removing {pdf_path} ...')

```

```

2629     path.unlink()
2630 time_1 = time.time()
2631 print(f'Persisting {pdf_path} ...')
2632 result = abjad.persist(part_file).as_pdf(pdf_path)
2633 print(result[0])
2634 print(result[1])
2635 print(result[2])
2636 success = result[3]
2637 if success is False:
2638     print('LilyPond failed!')
2639 time_2 = time.time()
2640 total_time = time_2 - time_1
2641 print(f'Total time: {total_time} seconds')
2642 if path.exists():
2643     print(f'Opening {pdf_path} ...')
2644     os.system(f'open {pdf_path}')
2645 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/
Section_I.ly').readlines()
2646 open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1/Section_I.ly', 'w'
).writelines(part_lines[15:-1])
2647
2648 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
2649 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2650 signature_copy = abjad.mutate(signatures).copy()
2651 staff_copy = abjad.mutate(staff).copy()
2652 part = abjad.Score()
2653 part.insert(0, staff)
2654 part.insert(0, signature_copy)
2655 part_file = abjad.LilyPondFile.new(
2656     part,
2657     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2658     )
2659 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2'
2660 pdf_path = f'{directory}/Section_I.pdf'
2661 path = pathlib.Path('Section_I.pdf')
2662 if path.exists():
2663     print(f'Removing {pdf_path} ...')
2664     path.unlink()
2665 time_1 = time.time()
2666 print(f'Persisting {pdf_path} ...')
2667 result = abjad.persist(part_file).as_pdf(pdf_path)
2668 print(result[0])
2669 print(result[1])
2670 print(result[2])
2671 success = result[3]
2672 if success is False:
2673     print('LilyPond failed!')
2674 time_2 = time.time()
2675 total_time = time_2 - time_1
2676 print(f'Total time: {total_time} seconds')
2677 if path.exists():
2678     print(f'Opening {pdf_path} ...')
2679     os.system(f'open {pdf_path}')
2680 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/
Section_I.ly').readlines()
2681 open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.)alto2/Section_I.ly', 'w'
).writelines(part_lines[15:-1])
2682

```

```

2683 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
2684     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2685     signature_copy = abjad.mutate(signatures).copy()
2686     staff_copy = abjad.mutate(staff).copy()
2687     part = abjad.Score()
2688     part.insert(0, staff)
2689     part.insert(0, signature_copy)
2690     part_file = abjad.LilyPondFile.new(
2691         part,
2692         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2693         _stylesheets/abjad.ily'],
2694         )
2695     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
2696     pdf_path = f'{directory}/Section_I.pdf'
2697     path = pathlib.Path('Section_I.pdf')
2698     if path.exists():
2699         print(f'Removing {pdf_path} ...')
2700         path.unlink()
2701     time_1 = time.time()
2702     print(f'Persisting {pdf_path} ...')
2703     result = abjad.persist(part_file).as_pdf(pdf_path)
2704     print(result[0])
2705     print(result[1])
2706     print(result[2])
2707     success = result[3]
2708     if success is False:
2709         print('LilyPond failed!')
2710     time_2 = time.time()
2711     total_time = time_2 - time_1
2712     print(f'Total time: {total_time} seconds')
2713     if path.exists():
2714         print(f'Opening {pdf_path} ...')
2715         os.system(f'open {pdf_path}')
2716     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/
2717     Section_I.ly').readlines()
2718     open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/Section_I.ly', 'w'
2719     .writelines(part_lines[15:-1])
2720
2721 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
2722     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2723     signature_copy = abjad.mutate(signatures).copy()
2724     staff_copy = abjad.mutate(staff).copy()
2725     part = abjad.Score()
2726     part.insert(0, staff)
2727     part.insert(0, signature_copy)
2728     part_file = abjad.LilyPondFile.new(
2729         part,
2730         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2731         _stylesheets/abjad.ily'],
2732         )
2733     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
2734     pdf_path = f'{directory}/Section_I.pdf'
2735     path = pathlib.Path('Section_I.pdf')
2736     if path.exists():
2737         print(f'Removing {pdf_path} ...')
2738         path.unlink()
2739     time_1 = time.time()
2740     print(f'Persisting {pdf_path} ...')
2741     result = abjad.persist(part_file).as_pdf(pdf_path)

```

```

2738 print(result[0])
2739 print(result[1])
2740 print(result[2])
2741 success = result[3]
2742 if success is False:
2743     print('LilyPond failed!')
2744 time_2 = time.time()
2745 total_time = time_2 - time_1
2746 print(f'Total time: {total_time} seconds')
2747 if path.exists():
2748     print(f'Opening {pdf_path} ...')
2749     os.system(f'open {pdf_path}')
2750 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/
Section_I.ly').readlines()
2751 open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.')alto4/Section_I.ly', 'w'
).writelines(part_lines[15:-1])
2752
2753 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
2754     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2755     signature_copy = abjad.mutate(signatures).copy()
2756     staff_copy = abjad.mutate(staff).copy()
2757     part = abjad.Score()
2758     part.insert(0, staff)
2759     part.insert(0, signature_copy)
2760     part_file = abjad.LilyPondFile.new(
2761         part,
2762         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2763         )
2764     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5'
2765     pdf_path = f'{directory}/Section_I.pdf'
2766     path = pathlib.Path('Section_I.pdf')
2767     if path.exists():
2768         print(f'Removing {pdf_path} ...')
2769         path.unlink()
2770     time_1 = time.time()
2771     print(f'Persisting {pdf_path} ...')
2772     result = abjad.persist(part_file).as_pdf(pdf_path)
2773     print(result[0])
2774     print(result[1])
2775     print(result[2])
2776     success = result[3]
2777     if success is False:
2778         print('LilyPond failed!')
2779     time_2 = time.time()
2780     total_time = time_2 - time_1
2781     print(f'Total time: {total_time} seconds')
2782     if path.exists():
2783         print(f'Opening {pdf_path} ...')
2784         os.system(f'open {pdf_path}')
2785     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/
Section_I.ly').readlines()
2786     open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.')alto5/Section_I.ly', 'w'
).writelines(part_lines[15:-1])
2787
2788 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
2789     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2790     signature_copy = abjad.mutate(signatures).copy()
2791     staff_copy = abjad.mutate(staff).copy()

```

```

2792 part = abjad.Score()
2793 part.insert(0, staff)
2794 part.insert(0, signature_copy)
2795 part_file = abjad.LilyPondFile.new(
2796     part,
2797     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2798 _stylesheets/abjad.ily'],
2799     )
2800 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
2801 pdf_path = f'{directory}/Section_I.pdf'
2802 path = pathlib.Path('Section_I.pdf')
2803 if path.exists():
2804     print(f'Removing {pdf_path} ...')
2805     path.unlink()
2806 time_1 = time.time()
2807 print(f'Persisting {pdf_path} ...')
2808 result = abjad.persist(part_file).as_pdf(pdf_path)
2809 print(result[0])
2810 print(result[1])
2811 print(result[2])
2812 success = result[3]
2813 if success is False:
2814     print('LilyPond failed!')
2815 time_2 = time.time()
2816 total_time = time_2 - time_1
2817 print(f'Total time: {total_time} seconds')
2818 if path.exists():
2819     print(f'Opening {pdf_path} ...')
2820     os.system(f'open {pdf_path}')
2821 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
2822 Section_I.ly').readlines()
2823 open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_I.ly', 'w'
2824 .writelines(part_lines[15:-1])
2825
2826 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
2827     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2828     signature_copy = abjad.mutate(signatures).copy()
2829     staff_copy = abjad.mutate(staff).copy()
2830     part = abjad.Score()
2831     part.insert(0, staff)
2832     part.insert(0, signature_copy)
2833     part_file = abjad.LilyPondFile.new(
2834         part,
2835         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2836 _stylesheets/abjad.ily'],
2837         )
2838 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
2839 pdf_path = f'{directory}/Section_I.pdf'
2840 path = pathlib.Path('Section_I.pdf')
2841 if path.exists():
2842     print(f'Removing {pdf_path} ...')
2843     path.unlink()
2844 time_1 = time.time()
2845 print(f'Persisting {pdf_path} ...')
2846 result = abjad.persist(part_file).as_pdf(pdf_path)
2847 print(result[0])
2848 print(result[1])
2849 print(result[2])
2850 success = result[3]
```

```

2847     if success is False:
2848         print('LilyPond failed!')
2849     time_2 = time.time()
2850     total_time = time_2 - time_1
2851     print(f'Total time: {total_time} seconds')
2852     if path.exists():
2853         print(f'Opening {pdf_path} ...')
2854         os.system(f'open {pdf_path}')
2855     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
Section_I.ly').readlines()
2856     open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Section_I.ly', ,
w').writelines(part_lines[15:-1])
2857
2858 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
2859     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2860     signature_copy = abjad.mutate(signatures).copy()
2861     staff_copy = abjad.mutate(staff).copy()
2862     part = abjad.Score()
2863     part.insert(0, staff)
2864     part.insert(0, signature_copy)
2865     part_file = abjad.LilyPondFile.new(
2866         part,
2867         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2868         )
2869     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
2870     pdf_path = f'{directory}/Section_I.pdf'
2871     path = pathlib.Path('Section_I.pdf')
2872     if path.exists():
2873         print(f'Removing {pdf_path} ...')
2874         path.unlink()
2875     time_1 = time.time()
2876     print(f'Persisting {pdf_path} ...')
2877     result = abjad.persist(part_file).as_pdf(pdf_path)
2878     print(result[0])
2879     print(result[1])
2880     print(result[2])
2881     success = result[3]
2882     if success is False:
2883         print('LilyPond failed!')
2884     time_2 = time.time()
2885     total_time = time_2 - time_1
2886     print(f'Total time: {total_time} seconds')
2887     if path.exists():
2888         print(f'Opening {pdf_path} ...')
2889         os.system(f'open {pdf_path}')
2890     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/
Section_I.ly').readlines()
2891     open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/Section_I.ly', ,
w').writelines(part_lines[15:-1])
2892
2893 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
2894     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2895     signature_copy = abjad.mutate(signatures).copy()
2896     staff_copy = abjad.mutate(staff).copy()
2897     part = abjad.Score()
2898     part.insert(0, staff)
2899     part.insert(0, signature_copy)
2900     part_file = abjad.LilyPondFile.new(

```

```

2901     part,
2902     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2903 _stylesheets/abjad.ily'],
2904     )
2905 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3'
2906 pdf_path = f'{directory}/Section_I.pdf'
2907 path = pathlib.Path('Section_I.pdf')
2908 if path.exists():
2909     print(f'Removing {pdf_path} ...')
2910     path.unlink()
2911 time_1 = time.time()
2912 print(f'Persisting {pdf_path} ...')
2913 result = abjad.persist(part_file).as_pdf(pdf_path)
2914 print(result[0])
2915 print(result[1])
2916 print(result[2])
2917 success = result[3]
2918 if success is False:
2919     print('LilyPond failed!')
2920 time_2 = time.time()
2921 total_time = time_2 - time_1
2922 print(f'Total time: {total_time} seconds')
2923 if path.exists():
2924     print(f'Opening {pdf_path} ...')
2925     os.system(f'open {pdf_path}')
2926 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3/
2927 Section_I.ly').readlines()
2928 open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3/Section_I.ly', 'w').writelines(part_lines[15:-1])
2929
2930 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
2931     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2932     signature_copy = abjad.mutate(signatures).copy()
2933     staff_copy = abjad.mutate(staff).copy()
2934     part = abjad.Score()
2935     part.insert(0, staff)
2936     part.insert(0, signature_copy)
2937     part_file = abjad.LilyPondFile.new(
2938         part,
2939         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2940 _stylesheets/abjad.ily'],
2941         )
2942 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.)tenor4'
2943 pdf_path = f'{directory}/Section_I.pdf'
2944 path = pathlib.Path('Section_I.pdf')
2945 if path.exists():
2946     print(f'Removing {pdf_path} ...')
2947     path.unlink()
2948 time_1 = time.time()
2949 print(f'Persisting {pdf_path} ...')
2950 result = abjad.persist(part_file).as_pdf(pdf_path)
2951 print(result[0])
2952 print(result[1])
2953 print(result[2])
2954 success = result[3]
2955 if success is False:
2956     print('LilyPond failed!')
2957 time_2 = time.time()
2958 total_time = time_2 - time_1

```

```

2956     print(f'Total time: {total_time} seconds')
2957     if path.exists():
2958         print(f'Opening {pdf_path} ...')
2959         os.system(f'open {pdf_path}')
2960     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
2961     Section_I.ly').readlines()
2962     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_I.ly', ,
2963     w').writelines(part_lines[15:-1])
2964
2965 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
2966     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2967     signature_copy = abjad.mutate(signatures).copy()
2968     staff_copy = abjad.mutate(staff).copy()
2969     part = abjad.Score()
2970     part.insert(0, staff)
2971     part.insert(0, signature_copy)
2972     part_file = abjad.LilyPondFile.new(
2973         part,
2974         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2975         _stylesheets/abjad.ily'],
2976         )
2977     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5'
2978     pdf_path = f'{directory}/Section_I.pdf'
2979     path = pathlib.Path('Section_I.pdf')
2980     if path.exists():
2981         print(f'Removing {pdf_path} ...')
2982         path.unlink()
2983     time_1 = time.time()
2984     print(f'Persisting {pdf_path} ...')
2985     result = abjad.persist(part_file).as_pdf(pdf_path)
2986     print(result[0])
2987     print(result[1])
2988     print(result[2])
2989     success = result[3]
2990     if success is False:
2991         print('LilyPond failed!')
2992     time_2 = time.time()
2993     total_time = time_2 - time_1
2994     print(f'Total time: {total_time} seconds')
2995     if path.exists():
2996         print(f'Opening {pdf_path} ...')
2997         os.system(f'open {pdf_path}')
2998     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/
2999     Section_I.ly').readlines()
3000     open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/Section_I.ly', ,
3001     w').writelines(part_lines[15:-1])
3002
3003 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
3004     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3005     signature_copy = abjad.mutate(signatures).copy()
3006     staff_copy = abjad.mutate(staff).copy()
3007     part = abjad.Score()
3008     part.insert(0, staff)

```

```

3009 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1'
3010 pdf_path = f'{directory}/Section_I.pdf'
3011 path = pathlib.Path('Section_I.pdf')
3012 if path.exists():
3013     print(f'Removing {pdf_path} ...')
3014     path.unlink()
3015 time_1 = time.time()
3016 print(f'Persisting {pdf_path} ...')
3017 result = abjad.persist(part_file).as_pdf(pdf_path)
3018 print(result[0])
3019 print(result[1])
3020 print(result[2])
3021 success = result[3]
3022 if success is False:
3023     print('LilyPond failed!')
3024 time_2 = time.time()
3025 total_time = time_2 - time_1
3026 print(f'Total time: {total_time} seconds')
3027 if path.exists():
3028     print(f'Opening {pdf_path} ...')
3029     os.system(f'open {pdf_path}')
3030 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/
3031 Section_I.ly').readlines()
3032 open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.)baritone1/Section_I.ly',
3033     'w').writelines(part_lines[15:-1])
3034
3035 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
3036     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3037     signature_copy = abjad.mutate(signatures).copy()
3038     staff_copy = abjad.mutate(staff).copy()
3039     part = abjad.Score()
3040     part.insert(0, staff)
3041     part.insert(0, signature_copy)
3042     part_file = abjad.LilyPondFile.new(
3043         part,
3044         includes=['first_stylesheet.ly', '/Users/evansdsg2/abjad/docs/source/
3045         _stylesheets/abjad.ly'],
3046         )
3047 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2'
3048 pdf_path = f'{directory}/Section_I.pdf'
3049 path = pathlib.Path('Section_I.pdf')
3050 if path.exists():
3051     print(f'Removing {pdf_path} ...')
3052     path.unlink()
3053 time_1 = time.time()
3054 print(f'Persisting {pdf_path} ...')
3055 result = abjad.persist(part_file).as_pdf(pdf_path)
3056 print(result[0])
3057 print(result[1])
3058 print(result[2])
3059 success = result[3]
3060 if success is False:
3061     print('LilyPond failed!')
3062 time_2 = time.time()
3063 total_time = time_2 - time_1
3064 print(f'Total time: {total_time} seconds')
3065 if path.exists():
3066     print(f'Opening {pdf_path} ...')
3067     os.system(f'open {pdf_path}')

```

```

3065 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/
3066 Section_I.ly').readlines()
3067 open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.)baritone2/Section_I.ly',
3068 , 'w').writelines(part_lines[15:-1])
3069
3070 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
3071     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3072     signature_copy = abjad.mutate(signatures).copy()
3073     staff_copy = abjad.mutate(staff).copy()
3074     part = abjad.Score()
3075     part.insert(0, staff)
3076     part.insert(0, signature_copy)
3077     part_file = abjad.LilyPondFile.new(
3078         part,
3079         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3080 _stylesheets/abjad.ily'],
3081         )
3082     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3'
3083     pdf_path = f'{directory}/Section_I.pdf'
3084     path = pathlib.Path('Section_I.pdf')
3085     if path.exists():
3086         print(f'Removing {pdf_path} ...')
3087         path.unlink()
3088     time_1 = time.time()
3089     print(f'Persisting {pdf_path} ...')
3090     result = abjad.persist(part_file).as_pdf(pdf_path)
3091     print(result[0])
3092     print(result[1])
3093     print(result[2])
3094     success = result[3]
3095     if success is False:
3096         print('LilyPond failed!')
3097     time_2 = time.time()
3098     total_time = time_2 - time_1
3099     print(f'Total time: {total_time} seconds')
3100     if path.exists():
3101         print(f'Opening {pdf_path} ...')
3102         os.system(f'open {pdf_path}')
3103     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/
3104 Section_I.ly').readlines()
3105     open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/Section_I.ly',
3106 , 'w').writelines(part_lines[15:-1])
3107
3108 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
3109     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3110     signature_copy = abjad.mutate(signatures).copy()
3111     staff_copy = abjad.mutate(staff).copy()
3112     part = abjad.Score()
3113     part.insert(0, staff)
3114     part.insert(0, signature_copy)
3115     part_file = abjad.LilyPondFile.new(
3116         part,
3117         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3118 _stylesheets/abjad.ily'],
3119         )
3120     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
3121     pdf_path = f'{directory}/Section_I.pdf'
3122     path = pathlib.Path('Section_I.pdf')
3123     if path.exists():

```

```

3118     print(f'Removing {pdf_path} ...')
3119     path.unlink()
3120 time_1 = time.time()
3121 print(f'Persisting {pdf_path} ...')
3122 result = abjad.persist(part_file).as_pdf(pdf_path)
3123 print(result[0])
3124 print(result[1])
3125 print(result[2])
3126 success = result[3]
3127 if success is False:
3128     print('LilyPond failed!')
3129 time_2 = time.time()
3130 total_time = time_2 - time_1
3131 print(f'Total time: {total_time} seconds')
3132 if path.exists():
3133     print(f'Opening {pdf_path} ...')
3134     os.system(f'open {pdf_path}')
3135 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
Section_I.ly').readlines()
3136 open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_I.ly', 'w'
').writelines(part_lines[15:-1])
3137
3138 for staff in abjad.iterate(score['Staff 20']).components(abjad.Staff):
3139 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3140 signature_copy = abjad.mutate(signatures).copy()
3141 staff_copy = abjad.mutate(staff).copy()
3142 part = abjad.Score()
3143 part.insert(0, staff)
3144 part.insert(0, signature_copy)
3145 part_file = abjad.LilyPondFile.new(
3146     part,
3147     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
3148     )
3149 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2'
3150 pdf_path = f'{directory}/Section_I.pdf'
3151 path = pathlib.Path('Section_I.pdf')
3152 if path.exists():
3153     print(f'Removing {pdf_path} ...')
3154     path.unlink()
3155 time_1 = time.time()
3156 print(f'Persisting {pdf_path} ...')
3157 result = abjad.persist(part_file).as_pdf(pdf_path)
3158 print(result[0])
3159 print(result[1])
3160 print(result[2])
3161 success = result[3]
3162 if success is False:
3163     print('LilyPond failed!')
3164 time_2 = time.time()
3165 total_time = time_2 - time_1
3166 print(f'Total time: {total_time} seconds')
3167 if path.exists():
3168     print(f'Opening {pdf_path} ...')
3169     os.system(f'open {pdf_path}')
3170 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/
Section_I.ly').readlines()
3171 open('/Users/evansdsg2/Scores/guerrero/Build/parts/20.)bass2/Section_I.ly', 'w'
').writelines(part_lines[15:-1])

```

```

3172 for staff in abjad.iterate(score['Staff 21']).components(abjad.Staff):
3173     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
3174     signature_copy = abjad.mutate(signatures).copy()
3175     staff_copy = abjad.mutate(staff).copy()
3176     part = abjad.Score()
3177     part.insert(0, staff)
3178     part.insert(0, signature_copy)
3179     part_file = abjad.LilyPondFile.new(
3180         part,
3181         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
3182         _stylesheets/abjad.ily'],
3183         )
3184     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.')contrabass'
3185     pdf_path = f'{directory}/Section_I.pdf'
3186     path = pathlib.Path('Section_I.pdf')
3187     if path.exists():
3188         print(f'Removing {pdf_path} ...')
3189         path.unlink()
3190     time_1 = time.time()
3191     print(f'Persisting {pdf_path} ...')
3192     result = abjad.persist(part_file).as_pdf(pdf_path)
3193     print(result[0])
3194     print(result[1])
3195     print(result[2])
3196     success = result[3]
3197     if success is False:
3198         print('LilyPond failed!')
3199     time_2 = time.time()
3200     total_time = time_2 - time_1
3201     print(f'Total time: {total_time} seconds')
3202     if path.exists():
3203         print(f'Opening {pdf_path} ...')
3204         os.system(f'open {pdf_path}')
3205     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.')contrabass
3206     /Section_I.ly').readlines()
3207     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.')contrabass/Section_I.ly
3208     ', 'w').writelines(part_lines[15:-1])

```

Listing 3.10: Invocation Source Code

## 3.11 Section J

```

1 import abjad
2 import itertools
3 import os
4 import pathlib
5 import time
6 import abjadext.rmakers
7 from MusicMaker import MusicMaker
8 from AttachmentHandler import AttachmentHandler
9 from random import random
10 from random import seed
11
12 print('Interpreting file ...')
13
14 time_signatures = [

```

```

15     abjad.TimeSignature(pair) for pair in [
16         (4, 4), (4, 4), (4, 4), (4, 4), (4, 4),
17         (4, 4), (9, 8),
18     ]
19 ]
20
21 bounds = abjad.mathtools.cumulative_sums([_.duration for _ in time_signatures])
22
23 def cyc(lst):
24     count = 0
25     while True:
26         yield lst[count%len(lst)]
27         count += 1
28
29 def grouper(lst1, lst2):
30     def cyc(lst):
31         c = 0
32         while True:
33             yield lst[c%len(lst)]
34             c += 1
35     lst1 = cyc(lst1)
36     return [next(lst1) if i == 1 else [next(lst1) for _ in range(i)] for i in lst2
37 ]
38
39 def reduceMod(list_length, rw):
40     return [(x % list_length) for x in rw]
41
42 soprano_note = [27, 17, 8, 0, 11, ]
43 soprano_1_note = [22, 16, 13, ]
44 soprano_2_note = [16, 13, 14, 13, ]
45 soprano_3_note = [13, 12, 13, 16, 26 ]
46 alto_1_note = [20, 12, 23, ]
47 alto_2_note = [12, 1, 12, 23, ]
48 alto_3_note = [1, 23, 1, 12, ]
49 alto_4_note = [20, 12, 20, 23, ]
50 alto_5_note = [12, 20, 1, 23, ]
51 alto_6_note = [1, 20, 12, ]
52 tenor_1_note = [17, -1, 25, 6,]
53 tenor_2_note = [6, 25, 17,]
54 tenor_3_note = [-1, 6, 17, 25, ]
55 tenor_4_note = [17, -1, 6, ]
56 tenor_5_note = [6, 25, 17, ]
57 baritone_1_note = [13, 24, 13, 6, ]
58 baritone_2_note = [6, 24, 13, 4, ]
59 bass_1_note = [9, 11, 11, 9, 11, ]
60 bass_2_note = [11, 9, 9, 11, 9, ]
61 contrabass_note = [-2, 2, 7, -2, 2, 7, 2, -2, ]
62
63 # -3 at bottom of chord for completion
64 soprano_chord = [17, 27, 11, 0, 8,]
65 soprano_1_chord = [22, [13.25, 16, 26.25, ], 16, [13.25, 16, 26.25, ], 13, [13.25,
66     16, 26.25, ], ]
67 soprano_2_chord = [16, [13, 14.75, 26.25, ], 13, [13, 14.75, 26.25, ], 14, [13,
68     14.75, 26.25, ], 13, [13, 14.75, 26.25, ], ] #maybe it's 13.25?
69 soprano_3_chord = [13, [12.75, 15.5, 26, ], 13, 12, [12.75, 15.5, 26, ], 16,
70     [12.75, 15.5, 26, ], 26 ]
71 alto_1_chord = [20, [12.5, 19, 27.75, 34, ], 12, [12.5, 19, 27.75, 34, ], 23, ]
72 alto_2_chord = [12, [12.5, 15.25, 25.5, ], 1, [12.5, 15.25, 25.5, ], 12, [12.5,
73     15.25, 25.5, ], 23 ]

```

```

    15.25, 25.5, ], 23, ]
70 alto_3_chord = [1, [1.75, 13.5, 22.25, 27, 30, ], 23, [1.75, 13.5, 22.25, 27, 30,
    ], 1, [1.75, 13.5, 22.25, 27, 30, ], 12, [1.75, 13.5, 22.25, 27, 30, ], ]
71 alto_4_chord = [20, [12.5, 15.25, 25.5, ], 12, [12.5, 15.25, 25.5, ], 20, [12.5,
    15.25, 25.5, ], 23, ]
72 alto_5_chord = [12, [1.75, 13.5, 22.25, 27, 30, ], 20, 1, [1.75, 13.5, 22.25, 27,
    30, ], 23, [1.75, 13.5, 22.25, 27, 30, ], ]
73 alto_6_chord = [1, [12.5, 19, 27.75, 34, ], 20, 12, [12.5, 19, 27.75, 34, ], ]
74 tenor_1_chord = [17, [6, 17.5, ], -1, 25, [6, 17.5, ], 6, [6, 17.5, ], ]
75 tenor_2_chord = [6, [6, 17.5, 25.5, 30, ], 25, [6, 17.5, 25.5, 30, ], 17, ]
76 tenor_3_chord = [-1, [6, 17.5, 25.5, 30.75, ], 6, [6, 17.5, 25.5, 30.75, ], 17,
    [6, 17.5, 25.5, 30.75, ], 25, [6, 17.5, 25.5, 30.75, ], ]
77 tenor_4_chord = [17, [6, 17.5, ], -1, [6, 17.5, ], 6, [6, 17.5, ], ]
78 tenor_5_chord = [6, [6, 17.5, 25.5, 30.75, ], 25, [6, 17.5, 25.5, 30.75, ], 17,
    [6, 17.5, 25.5, 30.75, ], ]
79 baritone_1_chord = [13, [13.25, 27.5, 33.75, ], 24, [13.25, 27.5, 33.75, ], 13,
    [13.25, 27.5, 33.75, ], 6, [13.25, 27.5, 33.75, ], ]
80 baritone_2_chord = [6, [4, 16.5, 23.5, ], 24, 13, [4, 16.5, 23.5, ], 4, ]
81 baritone_3_chord = [4, [7.75, 17.75, 25.5, 34, ], 13, [7.75, 17.75, 25.5, 34, ],
    6, [7.75, 17.75, 25.5, 34, ], 24, [7.75, 17.75, 25.5, 34, ], ]
82 bass_1_chord = [11, 18, 9, 11, 0, ]
83 bass_2_chord = [9, 11, 0, 11, 9, 18, ]
84 contrabass_chord = [-2, 7, 2, 16, 7, 18, 16, 25, 16, ]
85
86 def reduceMod(x, rw):
87     return [(y % x) for y in rw]
88
89 seed(22)
90 soprano_random_walk = []
91 soprano_random_walk.append(-1 if random() < 0.5 else 1)
92 for i in range(1, 1000):
93     movement = -1 if random() < 0.5 else 1
94     value = soprano_random_walk[i-1] + movement
95     soprano_random_walk.append(value)
96     soprano_walk_chord = [18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14,
        13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5, 6,
        7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ]
97 l = len(soprano_walk_chord)
98 soprano_random_walk_notes = [soprano_walk_chord[x] for x in reduceMod(l,
    soprano_random_walk)]
99
100 seed(23)
101 soprano_1_random_walk = []
102 soprano_1_random_walk.append(-1 if random() < 0.5 else 1)
103 for i in range(1, 1000):
104     movement = -1 if random() < 0.5 else 1
105     value = soprano_1_random_walk[i-1] + movement
106     soprano_1_random_walk.append(value)
107 soprano_1_walk_chord = [17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14,
        13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5, 6,
        7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ]
108 l = len(soprano_1_walk_chord)
109 soprano_1_random_walk_notes = [soprano_1_walk_chord[x] for x in reduceMod(l,
    soprano_1_random_walk)]
110
111 seed(24)
112 soprano_2_random_walk = []
113 soprano_2_random_walk.append(-1 if random() < 0.5 else 1)
114 for i in range(1, 1000):

```

```

115     movement = -1 if random() < 0.5 else 1
116     value = soprano_2_random_walk[i-1] + movement
117     soprano_2_random_walk.append(value)
118 soprano_2_random_walk.append(value)
119 soprano_2_walk_chord = [16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15,
120     14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, 4, 5,
121     6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ]
120 l = len(soprano_2_walk_chord)
121 soprano_2_random_walk_notes = [soprano_2_walk_chord[x] for x in reduceMod(l,
122     soprano_2_random_walk)]
122
123 seed(25)
124 soprano_3_random_walk = []
125 soprano_3_random_walk.append(-1 if random() < 0.5 else 1)
126 for i in range(1, 1000):
127     movement = -1 if random() < 0.5 else 1
128     value = soprano_3_random_walk[i-1] + movement
129     soprano_3_random_walk.append(value)
130 soprano_3_random_walk.append(value)
131 soprano_3_walk_chord = [15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16,
132     15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3,
133     4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ]
132 l = len(soprano_3_walk_chord)
133 soprano_3_random_walk_notes = [soprano_3_walk_chord[x] for x in reduceMod(l,
134     soprano_3_random_walk)]
134
135 seed(26)
136 alto_1_random_walk = []
137 alto_1_random_walk.append(-1 if random() < 0.5 else 1)
138 for i in range(1, 1000):
139     movement = -1 if random() < 0.5 else 1
140     value = alto_1_random_walk[i-1] + movement
141     alto_1_random_walk.append(value)
142 alto_1_walk_chord = [14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16,
143     15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3,
144     4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ]
143 l = len(alto_1_walk_chord)
144 alto_1_random_walk_notes = [alto_1_walk_chord[x] for x in reduceMod(l,
145     alto_1_random_walk)]
145
146 seed(27)
147 alto_2_random_walk = []
148 alto_2_random_walk.append(-1 if random() < 0.5 else 1)
149 for i in range(1, 1000):
150     movement = -1 if random() < 0.5 else 1
151     value = alto_2_random_walk[i-1] + movement
152     alto_2_random_walk.append(value)
153 alto_2_walk_chord = [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17,
154     16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2,
155     3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ]
154 l = len(alto_2_walk_chord)
155 alto_2_random_walk_notes = [alto_2_walk_chord[x] for x in reduceMod(l,
156     alto_2_random_walk)]
156
157 seed(28)
158 alto_3_random_walk = []
159 alto_3_random_walk.append(-1 if random() < 0.5 else 1)
160 for i in range(1, 1000):
161     movement = -1 if random() < 0.5 else 1

```

```

162     value = alto_3_random_walk[i-1] + movement
163     alto_3_random_walk.append(value)
164 alto_3_walk_chord = [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18,
165     17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1,
166     2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ]
167 l = len(alto_3_walk_chord)
168 alto_3_random_walk_notes = [alto_3_walk_chord[x] for x in reduceMod(l,
169     alto_3_random_walk)]
170
171 seed(29)
172 alto_4_random_walk = []
173 alto_4_random_walk.append(-1 if random() < 0.5 else 1)
174 for i in range(1, 1000):
175     movement = -1 if random() < 0.5 else 1
176     value = alto_4_random_walk[i-1] + movement
177     alto_4_random_walk.append(value)
178 alto_4_walk_chord = [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19,
179     18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1,
180     0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ]
181 l = len(alto_4_walk_chord)
182 alto_4_random_walk_notes = [alto_4_walk_chord[x] for x in reduceMod(l,
183     alto_4_random_walk)]
184
185 seed(30)
186 alto_5_random_walk = []
187 alto_5_random_walk.append(-1 if random() < 0.5 else 1)
188 for i in range(1, 1000):
189     movement = -1 if random() < 0.5 else 1
190     value = alto_5_random_walk[i-1] + movement
191     alto_5_random_walk.append(value)
192 alto_5_walk_chord = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21, 20,
193     19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2,
194     -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ]
195 l = len(alto_5_walk_chord)
196 alto_5_random_walk_notes = [alto_5_walk_chord[x] for x in reduceMod(l,
197     alto_5_random_walk)]
198
199 seed(31)
200 alto_6_random_walk = []
201 alto_6_random_walk.append(-1 if random() < 0.5 else 1)
202 for i in range(1, 1000):
203     movement = -1 if random() < 0.5 else 1
204     value = alto_6_random_walk[i-1] + movement
205     alto_6_random_walk.append(value)
206 alto_6_walk_chord = [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 21,
207     20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1,
208     -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, ]
209 l = len(alto_6_walk_chord)
210 alto_6_random_walk_notes = [alto_6_walk_chord[x] for x in reduceMod(l,
211     alto_6_random_walk)]
212
213 seed(32)
214 tenor_1_random_walk = []
215 tenor_1_random_walk.append(-1 if random() < 0.5 else 1)
216 for i in range(1, 1000):
217     movement = -1 if random() < 0.5 else 1
218     value = tenor_1_random_walk[i-1] + movement
219     tenor_1_random_walk.append(value)
220 tenor_1_walk_chord = [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
```

```

    21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
    -1, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, ]
209 l = len(tenor_1_walk_chord)
210 tenor_1_random_walk_notes = [tenor_1_walk_chord[x] for x in reduceMod(1,
    tenor_1_random_walk)]
211
212 seed(33)
213 tenor_2_random_walk = []
214 tenor_2_random_walk.append(-1 if random() < 0.5 else 1)
215 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = tenor_2_random_walk[i-1] + movement
    tenor_2_random_walk.append(value)
216 tenor_2_walk_chord = [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
    -1, -2, -1, 0, 1, 2, 3, 4, 5, 6, ]
217 l = len(tenor_2_walk_chord)
218 tenor_2_random_walk_notes = [tenor_2_walk_chord[x] for x in reduceMod(1,
    tenor_2_random_walk)]
219
220 seed(34)
221 tenor_3_random_walk = []
222 tenor_3_random_walk.append(-1 if random() < 0.5 else 1)
223 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = tenor_3_random_walk[i-1] + movement
    tenor_3_random_walk.append(value)
224 tenor_3_walk_chord = [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
    22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1,
    0, -1, -2, -1, 0, 1, 2, 3, 4, 5, ]
225 l = len(tenor_3_walk_chord)
226 tenor_3_random_walk_notes = [tenor_3_walk_chord[x] for x in reduceMod(1,
    tenor_3_random_walk)]
227
228 seed(35)
229 tenor_4_random_walk = []
230 tenor_4_random_walk.append(-1 if random() < 0.5 else 1)
231 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = tenor_4_random_walk[i-1] + movement
    tenor_4_random_walk.append(value)
232 tenor_4_walk_chord = [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
    21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2,
    1, 0, -1, -2, -1, 0, 1, 2, 3, 4, ]
233 l = len(tenor_4_walk_chord)
234 tenor_4_random_walk_notes = [tenor_4_walk_chord[x] for x in reduceMod(1,
    tenor_4_random_walk)]
235
236 seed(36)
237 tenor_5_random_walk = []
238 tenor_5_random_walk.append(-1 if random() < 0.5 else 1)
239 for i in range(1, 1000):
    movement = -1 if random() < 0.5 else 1
    value = tenor_5_random_walk[i-1] + movement
    tenor_5_random_walk.append(value)
240 tenor_5_walk_chord = [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
    20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4,
    3, 2, 1, 0, -1, -2, -1, 0, 1, 2, 3, ]
241 l = len(tenor_5_walk_chord)

```

```

254 tenor_5_random_walk_notes = [tenor_5_walk_chord[x] for x in reduceMod(1,
255   tenor_5_random_walk)]
256
256 seed(37)
257 baritone_1_random_walk = []
258 baritone_1_random_walk.append(-1 if random() < 0.5 else 1)
259 for i in range(1, 1000):
260     movement = -1 if random() < 0.5 else 1
261     value = baritone_1_random_walk[i-1] + movement
262     baritone_1_random_walk.append(value)
263 baritone_1_walk_chord = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
264   19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5,
265   4, 3, 2, 1, 0, -1, -2, -1, 0, 1, 2, ]
264 l = len(baritone_1_walk_chord)
265 baritone_1_random_walk_notes = [baritone_1_walk_chord[x] for x in reduceMod(l,
266   baritone_1_random_walk)]
266
267 seed(38)
268 baritone_2_random_walk = []
269 baritone_2_random_walk.append(-1 if random() < 0.5 else 1)
270 for i in range(1, 1000):
271     movement = -1 if random() < 0.5 else 1
272     value = baritone_2_random_walk[i-1] + movement
273     baritone_2_random_walk.append(value)
274 baritone_2_walk_chord = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
275   18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,
275   5, 4, 3, 2, 1, 0, -1, -2, -1, 0, 1, ]
275 l = len(baritone_2_walk_chord)
276 baritone_2_random_walk_notes = [baritone_2_walk_chord[x] for x in reduceMod(l,
277   baritone_2_random_walk)]
277
278 seed(39)
279 baritone_3_random_walk = []
280 baritone_3_random_walk.append(-1 if random() < 0.5 else 1)
281 for i in range(1, 1000):
282     movement = -1 if random() < 0.5 else 1
283     value = baritone_3_random_walk[i-1] + movement
284     baritone_3_random_walk.append(value)
285 baritone_3_walk_chord = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
286   17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8,
286   7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -1, 0, ]
286 l = len(baritone_3_walk_chord)
287 baritone_3_random_walk_notes = [baritone_3_walk_chord[x] for x in reduceMod(l,
288   baritone_3_random_walk)]
288
289 seed(40)
290 bass_1_random_walk = []
291 bass_1_random_walk.append(-1 if random() < 0.5 else 1)
292 for i in range(1, 1000):
293     movement = -1 if random() < 0.5 else 1
294     value = bass_1_random_walk[i-1] + movement
295     bass_1_random_walk.append(value)
296 bass_1_walk_chord = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
297   18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7,
297   6, 5, 4, 3, 2, 1, 0, -1, -2, -1, ]
297 l = len(bass_1_walk_chord)
298 bass_1_random_walk_notes = [bass_1_walk_chord[x] for x in reduceMod(l,
299   bass_1_random_walk)]
299

```

```

300 seed(41)
301 bass_2_random_walk = []
302 bass_2_random_walk.append(-1 if random() < 0.5 else 1)
303 for i in range(1, 1000):
304     movement = -1 if random() < 0.5 else 1
305     value = bass_2_random_walk[i-1] + movement
306     bass_2_random_walk.append(value)
307 bass_2_walk_chord = [-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
308     17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8,
309     7, 6, 5, 4, 3, 2, 1, 0, -1, -2, ]
310 l = len(bass_2_walk_chord)
311 bass_2_random_walk_notes = [bass_2_walk_chord[x] for x in reduceMod(l,
312     bass_2_random_walk)]
313
314 seed(42)
315 contrabass_random_walk = []
316 contrabass_random_walk.append(-1 if random() < 0.5 else 1)
317 for i in range(1, 1000):
318     movement = -1 if random() < 0.5 else 1
319     value = contrabass_random_walk[i-1] + movement
320     contrabass_random_walk.append(value)
321 contrabass_walk_chord = [-2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
322     15, 16, 17, 18, 19, 20, 21, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11,
323     10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, ]
324 l = len(contrabass_walk_chord)
325 contrabass_random_walk_notes = [contrabass_walk_chord[x] for x in reduceMod(l,
326     contrabass_random_walk)]
327
328 rmaker_one = abjadext.rmakers.TaleaRhythmMaker(
329     talea=abjadext.rmakers.Talea(
330         counts=[3, 2, 5, 2, 3, 1, 3, 2, 3, 1, 2, ],
331         denominator=8,
332         ),
333     beamSpecifier=abjadext.rmakers.BeamSpecifier(
334         beamDivisionsTogether=True,
335         beamRests=False,
336         ),
337     extraCountsPerDivision=[1, 0, -1, 1, -1, 0, 1, ],
338     logicalTieMasks=[
339         abjadext.rmakers.silence([3], 4),
340         ],
341     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
342         trivialize=True,
343         extractTrivial=True,
344         rewriteRestFilled=True,
345         rewriteSustained=True,
346         ),
347     )
348
349 rmaker_two = abjadext.rmakers.TaleaRhythmMaker(
350     talea=abjadext.rmakers.Talea(
351         counts=[1, 1, 3, 1, 5, 5, 3, 1, 1, 3, 1, 3, 3, 1, 3, 7, 1, 6, 1, 3, ],
352         denominator=8,
353         ),
354     beamSpecifier=abjadext.rmakers.BeamSpecifier(
355         beamDivisionsTogether=True,
356         beamRests=False,
357         ),
358     extraCountsPerDivision=[-1, 0, 1, -1, 1, 0, ],
359     )

```

```

353     logical_tie_masks=[
354         abjadext.rmakers.silence([4, 8, ], 9),
355         ],
356     tupletSpecifier=abjadext.rmakers.TupletSpecifier(
357         trivialize=True,
358         extract_trivial=True,
359         rewrite_rest_filled=True,
360         rewrite_sustained=True,
361         ),
362     )
363
364 attachmentHandler_one = AttachmentHandler(
365     starting_dynamic='f',
366     ending_dynamic='ff',
367     hairpin='<',
368     # articulation_list=['accent', '', 'tenuto', '', '', 'portato', '', '', '', ],
369 )
370
371 attachmentHandler_two = AttachmentHandler(
372     starting_dynamic='p',
373     ending_dynamic='mp',
374     hairpin='<',
375     articulation_list=['flageolet', 'flageolet', 'flageolet', 'flageolet', '',
376     'flageolet', 'halfopen', 'halfopen', 'halfopen', 'flageolet', 'stopped', '',
377     'stopped', 'stopped', 'halfopen', 'flageolet', 'halfopen', 'halfopen',
378     ', 'halfopen', 'stopped', 'stopped', 'stopped', ],
379 )
380
381 attachmentHandler_three = AttachmentHandler(
382     starting_dynamic='mf',
383     ending_dynamic='mp',
384     hairpin='>',
385 )
386
387 #####sopranino#####
388 sopranino_musicmaker_one = MusicMaker(
389     rmaker=rmaker_two,
390     pitches=sopranino_chord,
391     continuous=True,
392     attachment_handler=attachmentHandler_one,
393 )
394 sopranino_musicmaker_two = MusicMaker(
395     rmaker=rmaker_two,
396     pitches=sopranino_random_walk_notes,
397     continuous=True,
398     attachment_handler=attachmentHandler_two,
399 )
400 sopranino_musicmaker_three = MusicMaker(
401     rmaker=rmaker_two,
402     pitches=sopranino_note,
403     continuous=True,
404     attachment_handler=attachmentHandler_three,
405 )
406
407 #####soprano_one#####
408 soprano_one_musicmaker_one = MusicMaker(
409     rmaker=rmaker_two,
410     pitches=soprano_1_chord,
411     continuous=True,
412     attachment_handler=attachmentHandler_one,
413 )

```

```
409 )
410 soprano_one_musicmaker_two = MusicMaker(
411     rmaker=rmaker_two,
412     pitches=soprano_1_random_walk_notes,
413     continuous=True,
414     attachment_handler=attachment_handler_two,
415 )
416 soprano_one_musicmaker_three = MusicMaker(
417     rmaker=rmaker_two,
418     pitches=soprano_1_note,
419     continuous=True,
420     attachment_handler=attachment_handler_three,
421 )
422 #####soprano_two#####
423 soprano_two_musicmaker_one = MusicMaker(
424     rmaker=rmaker_two,
425     pitches=soprano_2_chord,
426     continuous=True,
427     attachment_handler=attachment_handler_one,
428 )
429 soprano_two_musicmaker_two = MusicMaker(
430     rmaker=rmaker_two,
431     pitches=soprano_2_random_walk_notes,
432     continuous=True,
433     attachment_handler=attachment_handler_two,
434 )
435 soprano_two_musicmaker_three = MusicMaker(
436     rmaker=rmaker_two,
437     pitches=soprano_2_note,
438     continuous=True,
439     attachment_handler=attachment_handler_three,
440 )
441 #####soprano_three#####
442 soprano_three_musicmaker_one = MusicMaker(
443     rmaker=rmaker_two,
444     pitches=soprano_3_chord,
445     continuous=True,
446     attachment_handler=attachment_handler_one,
447 )
448 soprano_three_musicmaker_two = MusicMaker(
449     rmaker=rmaker_two,
450     pitches=soprano_3_random_walk_notes,
451     continuous=True,
452     attachment_handler=attachment_handler_two,
453 )
454 soprano_three_musicmaker_three = MusicMaker(
455     rmaker=rmaker_two,
456     pitches=soprano_3_note,
457     continuous=True,
458     attachment_handler=attachment_handler_three,
459 )
460 #####alto_one#####
461 alto_one_musicmaker_one = MusicMaker(
462     rmaker=rmaker_two,
463     pitches=alto_1_chord,
464     continuous=True,
465     attachment_handler=attachment_handler_one,
466 )
467 alto_one_musicmaker_two = MusicMaker(
```

```
468     rmaker=rmaker_two,
469     pitches=alto_1_random_walk_notes,
470     continuous=True,
471     attachment_handler=attachment_handler_two,
472 )
473 alto_one_musicmaker_three = MusicMaker(
474     rmaker=rmaker_two,
475     pitches=alto_1_note,
476     continuous=True,
477     attachment_handler=attachment_handler_three,
478 )
479 #####alto_two#####
480 alto_two_musicmaker_one = MusicMaker(
481     rmaker=rmaker_two,
482     pitches=alto_2_chord,
483     continuous=True,
484     attachment_handler=attachment_handler_one,
485 )
486 alto_two_musicmaker_two = MusicMaker(
487     rmaker=rmaker_two,
488     pitches=alto_2_random_walk_notes,
489     continuous=True,
490     attachment_handler=attachment_handler_two,
491 )
492 alto_two_musicmaker_three = MusicMaker(
493     rmaker=rmaker_two,
494     pitches=alto_2_note,
495     continuous=True,
496     attachment_handler=attachment_handler_three,
497 )
498 #####alto_three#####
499 alto_three_musicmaker_one = MusicMaker(
500     rmaker=rmaker_two,
501     pitches=alto_3_chord,
502     continuous=True,
503     attachment_handler=attachment_handler_one,
504 )
505 alto_three_musicmaker_two = MusicMaker(
506     rmaker=rmaker_two,
507     pitches=alto_3_random_walk_notes,
508     continuous=True,
509     attachment_handler=attachment_handler_two,
510 )
511 alto_three_musicmaker_three = MusicMaker(
512     rmaker=rmaker_two,
513     pitches=alto_3_note,
514     continuous=True,
515     attachment_handler=attachment_handler_three,
516 )
517 #####alto_four#####
518 alto_four_musicmaker_one = MusicMaker(
519     rmaker=rmaker_two,
520     pitches=alto_4_chord,
521     continuous=True,
522     attachment_handler=attachment_handler_one,
523 )
524 alto_four_musicmaker_two = MusicMaker(
525     rmaker=rmaker_two,
526     pitches=alto_4_random_walk_notes,
```

```
527     continuous=True,
528     attachment_handler=attachment_handler_two,
529 )
530 alto_four_musicmaker_three = MusicMaker(
531     rmaker=rmaker_two,
532     pitches=alto_4_note,
533     continuous=True,
534     attachment_handler=attachment_handler_three,
535 )
536 #####alto_five#####
537 alto_five_musicmaker_one = MusicMaker(
538     rmaker=rmaker_two,
539     pitches=alto_5_chord,
540     continuous=True,
541     attachment_handler=attachment_handler_one,
542 )
543 alto_five_musicmaker_two = MusicMaker(
544     rmaker=rmaker_two,
545     pitches=alto_5_random_walk_notes,
546     continuous=True,
547     attachment_handler=attachment_handler_two,
548 )
549 alto_five_musicmaker_three = MusicMaker(
550     rmaker=rmaker_two,
551     pitches=alto_5_note,
552     continuous=True,
553     attachment_handler=attachment_handler_three,
554 )
555 #####alto_six#####
556 alto_six_musicmaker_one = MusicMaker(
557     rmaker=rmaker_two,
558     pitches=alto_6_chord,
559     continuous=True,
560     attachment_handler=attachment_handler_one,
561 )
562 alto_six_musicmaker_two = MusicMaker(
563     rmaker=rmaker_two,
564     pitches=alto_6_random_walk_notes,
565     continuous=True,
566     attachment_handler=attachment_handler_two,
567 )
568 alto_six_musicmaker_three = MusicMaker(
569     rmaker=rmaker_two,
570     pitches=alto_6_note,
571     continuous=True,
572     attachment_handler=attachment_handler_three,
573 )
574 #####tenor_one#####
575 tenor_one_musicmaker_one = MusicMaker(
576     rmaker=rmaker_two,
577     pitches=tenor_1_chord,
578     continuous=True,
579     attachment_handler=attachment_handler_one,
580 )
581 tenor_one_musicmaker_two = MusicMaker(
582     rmaker=rmaker_two,
583     pitches=tenor_1_random_walk_notes,
584     continuous=True,
585     attachment_handler=attachment_handler_two,
```

```
586 )
587 tenor_one_musicmaker_three = MusicMaker(
588     rmaker=rmaker_two,
589     pitches=tenor_1_note,
590     continuous=True,
591     attachment_handler=attachment_handler_three,
592 )
593 #####tenor_two#####
594 tenor_two_musicmaker_one = MusicMaker(
595     rmaker=rmaker_two,
596     pitches=tenor_2_chord,
597     continuous=True,
598     attachment_handler=attachment_handler_one,
599 )
600 tenor_two_musicmaker_two = MusicMaker(
601     rmaker=rmaker_two,
602     pitches=tenor_2_random_walk_notes,
603     continuous=True,
604     attachment_handler=attachment_handler_two,
605 )
606 tenor_two_musicmaker_three = MusicMaker(
607     rmaker=rmaker_two,
608     pitches=tenor_2_note,
609     continuous=True,
610     attachment_handler=attachment_handler_three,
611 )
612 #####tenor_three#####
613 tenor_three_musicmaker_one = MusicMaker(
614     rmaker=rmaker_two,
615     pitches=tenor_3_chord,
616     continuous=True,
617     attachment_handler=attachment_handler_one,
618 )
619 tenor_three_musicmaker_two = MusicMaker(
620     rmaker=rmaker_two,
621     pitches=tenor_3_random_walk_notes,
622     continuous=True,
623     attachment_handler=attachment_handler_two,
624 )
625 tenor_three_musicmaker_three = MusicMaker(
626     rmaker=rmaker_two,
627     pitches=tenor_3_note,
628     continuous=True,
629     attachment_handler=attachment_handler_three,
630 )
631 #####tenor_four#####
632 tenor_four_musicmaker_one = MusicMaker(
633     rmaker=rmaker_two,
634     pitches=tenor_4_chord,
635     continuous=True,
636     attachment_handler=attachment_handler_one,
637 )
638 tenor_four_musicmaker_two = MusicMaker(
639     rmaker=rmaker_two,
640     pitches=tenor_4_random_walk_notes,
641     continuous=True,
642     attachment_handler=attachment_handler_two,
643 )
644 tenor_four_musicmaker_three = MusicMaker(
```

```
645     rmaker=rmaker_two,
646     pitches=tenor_4_note,
647     continuous=True,
648     attachment_handler=attachment_handler_three,
649 )
650 #####tenor_five#####
651 tenor_five_musicmaker_one = MusicMaker(
652     rmaker=rmaker_two,
653     pitches=tenor_5_chord,
654     continuous=True,
655     attachment_handler=attachment_handler_one,
656 )
657 tenor_five_musicmaker_two = MusicMaker(
658     rmaker=rmaker_two,
659     pitches=tenor_5_random_walk_notes,
660     continuous=True,
661     attachment_handler=attachment_handler_two,
662 )
663 tenor_five_musicmaker_three = MusicMaker(
664     rmaker=rmaker_two,
665     pitches=tenor_5_note,
666     continuous=True,
667     attachment_handler=attachment_handler_three,
668 )
669 #####baritone_one#####
670 baritone_one_musicmaker_one = MusicMaker(
671     rmaker=rmaker_two,
672     pitches=baritone_1_chord,
673     continuous=True,
674     attachment_handler=attachment_handler_one,
675 )
676 baritone_one_musicmaker_two = MusicMaker(
677     rmaker=rmaker_two,
678     pitches=baritone_1_random_walk_notes,
679     continuous=True,
680     attachment_handler=attachment_handler_two,
681 )
682 baritone_one_musicmaker_three = MusicMaker(
683     rmaker=rmaker_two,
684     pitches=baritone_1_note,
685     continuous=True,
686     attachment_handler=attachment_handler_three,
687 )
688 #####baritone_two#####
689 baritone_two_musicmaker_one = MusicMaker(
690     rmaker=rmaker_two,
691     pitches=baritone_2_chord,
692     continuous=True,
693     attachment_handler=attachment_handler_one,
694 )
695 baritone_two_musicmaker_two = MusicMaker(
696     rmaker=rmaker_two,
697     pitches=baritone_2_random_walk_notes,
698     continuous=True,
699     attachment_handler=attachment_handler_two,
700 )
701 baritone_two_musicmaker_three = MusicMaker(
702     rmaker=rmaker_two,
703     pitches=baritone_2_note,
```

```
704     continuous=True,
705     attachment_handler=attachment_handler_three,
706 )
707 #####baritone_three#####
708 baritone_three_musicmaker_one = MusicMaker(
709     rmaker=rmaker_two,
710     pitches=baritone_3_chord,
711     continuous=True,
712     attachment_handler=attachment_handler_one,
713 )
714 baritone_three_musicmaker_two = MusicMaker(
715     rmaker=rmaker_two,
716     pitches=baritone_3_random_walk_notes,
717     continuous=True,
718     attachment_handler=attachment_handler_two,
719 )
720 baritone_three_musicmaker_three = MusicMaker(
721     rmaker=rmaker_two,
722     pitches=baritone_3_note,
723     continuous=True,
724     attachment_handler=attachment_handler_three,
725 )
726 #####bass_one#####
727 bass_one_musicmaker_one = MusicMaker(
728     rmaker=rmaker_two,
729     pitches=bass_1_chord,
730     continuous=True,
731     attachment_handler=attachment_handler_one,
732 )
733 bass_one_musicmaker_two = MusicMaker(
734     rmaker=rmaker_two,
735     pitches=bass_1_random_walk_notes,
736     continuous=True,
737     attachment_handler=attachment_handler_two,
738 )
739 bass_one_musicmaker_three = MusicMaker(
740     rmaker=rmaker_two,
741     pitches=bass_1_note,
742     continuous=True,
743     attachment_handler=attachment_handler_three,
744 )
745 #####bass_two#####
746 bass_two_musicmaker_one = MusicMaker(
747     rmaker=rmaker_two,
748     pitches=bass_2_chord,
749     continuous=True,
750     attachment_handler=attachment_handler_one,
751 )
752 bass_two_musicmaker_two = MusicMaker(
753     rmaker=rmaker_two,
754     pitches=bass_2_random_walk_notes,
755     continuous=True,
756     attachment_handler=attachment_handler_two,
757 )
758 bass_two_musicmaker_three = MusicMaker(
759     rmaker=rmaker_two,
760     pitches=bass_2_note,
761     continuous=True,
762     attachment_handler=attachment_handler_three,
```

```

763 )
764 #####contrabass#####
765 contrabass_musicmaker_one = MusicMaker(
766     rmaker=rmaker_two,
767     pitches=contrabass_chord,
768     continuous=True,
769     attachment_handler=attachment_handler_one,
770 )
771 contrabass_musicmaker_two = MusicMaker(
772     rmaker=rmaker_two,
773     pitches=contrabass_random_walk_notes,
774     continuous=True,
775     attachment_handler=attachment_handler_two,
776 )
777 contrabass_musicmaker_three = MusicMaker(
778     rmaker=rmaker_two,
779     pitches=contrabass_note,
780     continuous=True,
781     attachment_handler=attachment_handler_three,
782 )
783
784 silence_maker = abjadext.rmakers.NoteRhythmMaker(
785     division_masks=[
786         abjadext.rmakers.SilenceMask(
787             pattern=abjad.index([0], 1),
788         ),
789     ],
790 )
791
792 class MusicSpecifier:
793
794     def __init__(self, music_maker, voice_name):
795         self.music_maker = music_maker
796         self.voice_name = voice_name
797
798 print('Collecting timespans and rmakers ...')
799
800 voice_1_timespan_list = abjad.TimespanList([
801     abjad.AnnotatedTimespan(
802         start_offset=start_offset,
803         stop_offset=stop_offset,
804         annotation=MusicSpecifier(
805             music_maker=music_maker,
806             voice_name='Voice 1',
807         ),
808     )
809     for start_offset, stop_offset, music_maker in [
810         [(0, 8), (2, 8), soprano_musicmaker_two],
811         [(2, 8), (4, 8), soprano_musicmaker_one],
812         [(4, 8), (6, 8), soprano_musicmaker_two],
813         [(6, 8), (8, 8), soprano_musicmaker_two],
814         [(10, 8), (12, 8), soprano_musicmaker_two],
815         [(12, 8), (14, 8), soprano_musicmaker_two],
816         [(14, 8), (16, 8), soprano_musicmaker_two],
817         [(16, 8), (18, 8), soprano_musicmaker_one],
818         [(18, 8), (20, 8), soprano_musicmaker_two],
819         [(20, 8), (22, 8), soprano_musicmaker_two],
820         [(22, 8), (24, 8), soprano_musicmaker_two],
821     ]

```

```

822     [(26, 8), (28, 8), soprano_musicmaker_one],
823     [(28, 8), (30, 8), soprano_musicmaker_two],
824     [(30, 8), (32, 8), soprano_musicmaker_two],
825     [(32, 8), (34, 8), soprano_musicmaker_two],
826     [(34, 8), (36, 8), soprano_musicmaker_three],
827     [(36, 8), (38, 8), soprano_musicmaker_three],
828     [(38, 8), (40, 8), soprano_musicmaker_one],
829     [(40, 8), (42, 8), soprano_musicmaker_two],
830     [(42, 8), (44, 8), soprano_musicmaker_two],
831     [(44, 8), (46, 8), soprano_musicmaker_two],
832     [(46, 8), (48, 8), soprano_musicmaker_one],
833     [(48, 8), (50, 8), soprano_musicmaker_three],
834
835
836     [(52, 8), (54, 8), soprano_musicmaker_two],
837     [(54, 8), (56, 8), soprano_musicmaker_two],
838     [(56, 8), (57, 8), silence_maker],
839 ]
840 ])
841
842 voice_2_timestrap_list = abjad.TimespanList([
843     abjad.AnnotatedTimespan(
844         start_offset=start_offset,
845         stop_offset=stop_offset,
846         annotation=MusicSpecifier(
847             music_maker=music_maker,
848             voice_name='Voice 2',
849         ),
850     )
851     for start_offset, stop_offset, music_maker in [
852         [(0, 8), (2, 8), soprano_one_musicmaker_two],
853         [(2, 8), (4, 8), soprano_one_musicmaker_three],
854         [(4, 8), (6, 8), soprano_one_musicmaker_two],
855
856         [(8, 8), (10, 8), soprano_one_musicmaker_three],
857         [(10, 8), (12, 8), soprano_one_musicmaker_two],
858         [(12, 8), (14, 8), soprano_one_musicmaker_two],
859         [(14, 8), (16, 8), soprano_one_musicmaker_one],
860         [(16, 8), (18, 8), soprano_one_musicmaker_three],
861         [(18, 8), (20, 8), soprano_one_musicmaker_one],
862
863         [(22, 8), (24, 8), soprano_one_musicmaker_one],
864         [(24, 8), (26, 8), soprano_one_musicmaker_one],
865         [(26, 8), (28, 8), soprano_one_musicmaker_three],
866         [(28, 8), (30, 8), soprano_one_musicmaker_two],
867         [(30, 8), (32, 8), soprano_one_musicmaker_two],
868         [(32, 8), (34, 8), soprano_one_musicmaker_two],
869         [(34, 8), (36, 8), soprano_one_musicmaker_two],
870         [(36, 8), (38, 8), soprano_one_musicmaker_three],
871
872         [(40, 8), (42, 8), soprano_one_musicmaker_one],
873         [(42, 8), (44, 8), soprano_one_musicmaker_one],
874
875         [(46, 8), (48, 8), soprano_one_musicmaker_one],
876         [(48, 8), (50, 8), soprano_one_musicmaker_two],
877         [(50, 8), (52, 8), soprano_one_musicmaker_three],
878         [(52, 8), (54, 8), soprano_one_musicmaker_two],
879         [(54, 8), (56, 8), soprano_one_musicmaker_one],
880     ]

```

```

881 ])
882
883 voice_3_timespan_list = abjad.TimespanList([
884     abjad.AnnotatedTimespan(
885         start_offset=start_offset,
886         stop_offset=stop_offset,
887         annotation=MusicSpecifier(
888             music_maker=music_maker,
889             voice_name='Voice 3',
890         ),
891     )
892     for start_offset, stop_offset, music_maker in [
893         [(0, 8), (2, 8), soprano_two_musicmaker_one],
894         [(2, 8), (4, 8), soprano_two_musicmaker_one],
895         [(4, 8), (6, 8), soprano_two_musicmaker_one],
896         [(6, 8), (8, 8), soprano_two_musicmaker_one],
897         [(8, 8), (10, 8), soprano_two_musicmaker_one],
898         [(10, 8), (12, 8), soprano_two_musicmaker_three],
899         [(12, 8), (14, 8), soprano_two_musicmaker_one],
900         [(14, 8), (16, 8), soprano_two_musicmaker_one],
901         [(16, 8), (18, 8), soprano_two_musicmaker_two],
902
903         [(20, 8), (22, 8), soprano_two_musicmaker_two],
904         [(22, 8), (24, 8), soprano_two_musicmaker_three],
905         [(24, 8), (26, 8), soprano_two_musicmaker_two],
906         [(26, 8), (28, 8), soprano_two_musicmaker_two],
907         [(28, 8), (30, 8), soprano_two_musicmaker_two],
908         [(30, 8), (32, 8), soprano_two_musicmaker_one],
909         [(32, 8), (34, 8), soprano_two_musicmaker_one],
910         [(34, 8), (36, 8), soprano_two_musicmaker_one],
911         [(36, 8), (38, 8), soprano_two_musicmaker_three],
912         [(38, 8), (40, 8), soprano_two_musicmaker_one],
913         [(40, 8), (42, 8), soprano_two_musicmaker_one],
914         [(42, 8), (44, 8), soprano_two_musicmaker_two],
915         [(44, 8), (46, 8), soprano_two_musicmaker_three],
916         [(46, 8), (48, 8), soprano_two_musicmaker_two],
917         [(48, 8), (50, 8), soprano_two_musicmaker_two],
918         [(50, 8), (52, 8), soprano_two_musicmaker_one],
919     ]
920 ])
921
922 voice_4_timespan_list = abjad.TimespanList([
923     abjad.AnnotatedTimespan(
924         start_offset=start_offset,
925         stop_offset=stop_offset,
926         annotation=MusicSpecifier(
927             music_maker=music_maker,
928             voice_name='Voice 4',
929         ),
930     )
931     for start_offset, stop_offset, music_maker in [
932         [(0, 8), (2, 8), soprano_three_musicmaker_two],
933         [(2, 8), (4, 8), soprano_three_musicmaker_two],
934         [(4, 8), (6, 8), soprano_three_musicmaker_two],
935         [(6, 8), (8, 8), soprano_three_musicmaker_three],
936         [(8, 8), (10, 8), soprano_three_musicmaker_one],
937         [(10, 8), (12, 8), soprano_three_musicmaker_one],
938         [(12, 8), (14, 8), soprano_three_musicmaker_one],
939         [(14, 8), (16, 8), soprano_three_musicmaker_three],

```

```

940     [(16, 8), (18, 8), soprano_three_musicmaker_two],
941
942     [(20, 8), (22, 8), soprano_three_musicmaker_two],
943     [(22, 8), (24, 8), soprano_three_musicmaker_three],
944     [(24, 8), (26, 8), soprano_three_musicmaker_two],
945     [(26, 8), (28, 8), soprano_three_musicmaker_two],
946     [(28, 8), (30, 8), soprano_three_musicmaker_two],
947     [(30, 8), (32, 8), soprano_three_musicmaker_three],
948     [(32, 8), (34, 8), soprano_three_musicmaker_one],
949     [(34, 8), (36, 8), soprano_three_musicmaker_one],
950     [(36, 8), (38, 8), soprano_three_musicmaker_one],
951     [(39, 8), (40, 8), soprano_three_musicmaker_three],
952
953     [(42, 8), (44, 8), soprano_three_musicmaker_one],
954     [(44, 8), (46, 8), soprano_three_musicmaker_two],
955     [(46, 8), (48, 8), soprano_three_musicmaker_three],
956     [(48, 8), (50, 8), soprano_three_musicmaker_two],
957     [(50, 8), (52, 8), soprano_three_musicmaker_two],
958     [(52, 8), (54, 8), soprano_three_musicmaker_two],
959     [(54, 8), (56, 8), soprano_three_musicmaker_three],
960 ]
961 ])
962
963 voice_5_timestspan_list = abjad.TimespanList([
964     abjad.AnnotatedTimespan(
965         start_offset=start_offset,
966         stop_offset=stop_offset,
967         annotation=MusicSpecifier(
968             music_maker=music_maker,
969             voice_name='Voice 5',
970         ),
971     ),
972     for start_offset, stop_offset, music_maker in [
973         [(0, 8), (2, 8), alto_one_musicmaker_one],
974         [(2, 8), (4, 8), alto_one_musicmaker_one],
975         [(4, 8), (6, 8), alto_one_musicmaker_two],
976         [(6, 8), (8, 8), alto_one_musicmaker_two],
977         [(8, 8), (10, 8), alto_one_musicmaker_three],
978
979         [(12, 8), (14, 8), alto_one_musicmaker_two],
980         [(14, 8), (16, 8), alto_one_musicmaker_one],
981         [(16, 8), (18, 8), alto_one_musicmaker_one],
982         [(18, 8), (20, 8), alto_one_musicmaker_three],
983
984         [(22, 8), (24, 8), alto_one_musicmaker_one],
985         [(24, 8), (26, 8), alto_one_musicmaker_two],
986         [(26, 8), (28, 8), alto_one_musicmaker_two],
987         [(28, 8), (30, 8), alto_one_musicmaker_three],
988         [(30, 8), (32, 8), alto_one_musicmaker_two],
989         [(32, 8), (34, 8), alto_one_musicmaker_two],
990         [(34, 8), (36, 8), alto_one_musicmaker_one],
991
992         [(38, 8), (40, 8), alto_one_musicmaker_three],
993         [(40, 8), (42, 8), alto_one_musicmaker_one],
994         [(42, 8), (44, 8), alto_one_musicmaker_one],
995         [(44, 8), (46, 8), alto_one_musicmaker_two],
996         [(46, 8), (48, 8), alto_one_musicmaker_two],
997         [(48, 8), (50, 8), alto_one_musicmaker_three],
998         [(50, 8), (52, 8), alto_one_musicmaker_two],

```

```

999
1000     [(54, 8), (56, 8), alto_one_musicmaker_one],
1001 ]
1002 ])
1003
1004 voice_6_timespan_list = abjad.TimespanList([
1005     abjad.AnnotatedTimespan(
1006         start_offset=start_offset,
1007         stop_offset=stop_offset,
1008         annotation=MusicSpecifier(
1009             music_maker=music_maker,
1010             voice_name='Voice 6',
1011         ),
1012     ),
1013     for start_offset, stop_offset, music_maker in [
1014         [(0, 8), (2, 8), alto_two_musicmaker_one],
1015         [(2, 8), (4, 8), alto_two_musicmaker_one],
1016         [(4, 8), (6, 8), alto_two_musicmaker_one],
1017         [(6, 8), (8, 8), alto_two_musicmaker_one],
1018         [(8, 8), (10, 8), alto_two_musicmaker_two],
1019         [(10, 8), (12, 8), alto_two_musicmaker_three],
1020
1021         [(14, 8), (16, 8), alto_two_musicmaker_two],
1022         [(16, 8), (18, 8), alto_two_musicmaker_two],
1023         [(18, 8), (20, 8), alto_two_musicmaker_two],
1024         [(20, 8), (22, 8), alto_two_musicmaker_one],
1025         [(22, 8), (24, 8), alto_two_musicmaker_three],
1026         [(24, 8), (26, 8), alto_two_musicmaker_one],
1027
1028         [(28, 8), (30, 8), alto_two_musicmaker_one],
1029         [(30, 8), (32, 8), alto_two_musicmaker_two],
1030         [(32, 8), (34, 8), alto_two_musicmaker_two],
1031         [(34, 8), (36, 8), alto_two_musicmaker_three],
1032         [(36, 8), (38, 8), alto_two_musicmaker_two],
1033         [(38, 8), (40, 8), alto_two_musicmaker_one],
1034         [(40, 8), (42, 8), alto_two_musicmaker_one],
1035         [(42, 8), (44, 8), alto_two_musicmaker_one],
1036         [(44, 8), (46, 8), alto_two_musicmaker_two],
1037         [(46, 8), (48, 8), alto_two_musicmaker_three],
1038         [(48, 8), (50, 8), alto_two_musicmaker_one],
1039         [(50, 8), (52, 8), alto_two_musicmaker_two],
1040         [(52, 8), (54, 8), alto_two_musicmaker_one],
1041     ]
1042 ])
1043
1044 voice_7_timespan_list = abjad.TimespanList([
1045     abjad.AnnotatedTimespan(
1046         start_offset=start_offset,
1047         stop_offset=stop_offset,
1048         annotation=MusicSpecifier(
1049             music_maker=music_maker,
1050             voice_name='Voice 7',
1051         ),
1052     ),
1053     for start_offset, stop_offset, music_maker in [
1054         [(0, 8), (2, 8), alto_three_musicmaker_two],
1055
1056         [(4, 8), (6, 8), alto_three_musicmaker_two],
1057         [(6, 8), (8, 8), alto_three_musicmaker_two],

```

```

1058     [(8, 8), (10, 8), alto_three_musicmaker_two],
1059
1060     [(12, 8), (14, 8), alto_three_musicmaker_two],
1061     [(14, 8), (16, 8), alto_three_musicmaker_one],
1062     [(16, 8), (18, 8), alto_three_musicmaker_three],
1063     [(18, 8), (20, 8), alto_three_musicmaker_one],
1064
1065     [(22, 8), (24, 8), alto_three_musicmaker_two],
1066     [(24, 8), (26, 8), alto_three_musicmaker_two],
1067     [(26, 8), (28, 8), alto_three_musicmaker_two],
1068
1069     [(32, 8), (34, 8), alto_three_musicmaker_one],
1070     [(34, 8), (36, 8), alto_three_musicmaker_three],
1071     [(36, 8), (38, 8), alto_three_musicmaker_one],
1072     [(38, 8), (40, 8), alto_three_musicmaker_one],
1073     [(40, 8), (42, 8), alto_three_musicmaker_one],
1074
1075     [(44, 8), (46, 8), alto_three_musicmaker_two],
1076     [(46, 8), (48, 8), alto_three_musicmaker_two],
1077     [(48, 8), (50, 8), alto_three_musicmaker_two],
1078     [(50, 8), (52, 8), alto_three_musicmaker_two],
1079     [(52, 8), (54, 8), alto_three_musicmaker_three],
1080 ]
1081 ])
1082
1083 voice_8_timespan_list = abjad.TimespanList([
1084     abjad.AnnotatedTimespan(
1085         start_offset=start_offset,
1086         stop_offset=stop_offset,
1087         annotation=MusicSpecifier(
1088             music_maker=music_maker,
1089             voice_name='Voice 8',
1090         ),
1091     )
1092     for start_offset, stop_offset, music_maker in [
1093         [(0, 8), (2, 8), alto_four_musicmaker_two],
1094         [(2, 8), (4, 8), alto_four_musicmaker_two],
1095         [(4, 8), (6, 8), alto_four_musicmaker_two],
1096         [(6, 8), (8, 8), alto_four_musicmaker_one],
1097         [(8, 8), (10, 8), alto_four_musicmaker_one],
1098         [(10, 8), (12, 8), alto_four_musicmaker_one],
1099         [(12, 8), (14, 8), alto_four_musicmaker_two],
1100
1101         [(16, 8), (18, 8), alto_four_musicmaker_two],
1102         [(18, 8), (20, 8), alto_four_musicmaker_one],
1103         [(20, 8), (22, 8), alto_four_musicmaker_three],
1104         [(22, 8), (24, 8), alto_four_musicmaker_one],
1105         [(24, 8), (26, 8), alto_four_musicmaker_one],
1106         [(26, 8), (28, 8), alto_four_musicmaker_two],
1107         [(28, 8), (30, 8), alto_four_musicmaker_two],
1108
1109         [(32, 8), (34, 8), alto_four_musicmaker_two],
1110         [(34, 8), (36, 8), alto_four_musicmaker_one],
1111         [(36, 8), (38, 8), alto_four_musicmaker_one],
1112         [(38, 8), (39, 8), alto_four_musicmaker_one],
1113         [(40, 8), (42, 8), alto_four_musicmaker_one],
1114         [(42, 8), (44, 8), alto_four_musicmaker_three],
1115         [(44, 8), (46, 8), alto_four_musicmaker_two],
1116

```

```

1117     [(48, 8), (50, 8), alto_four_musicmaker_two],
1118     [(50, 8), (52, 8), alto_four_musicmaker_two],
1119     [(52, 8), (54, 8), alto_four_musicmaker_two],
1120     [(54, 8), (56, 8), alto_four_musicmaker_one],
1121 ]
1122 ])
1123
1124 voice_9_timespan_list = abjad.TimespanList([
1125     abjad.AnnotatedTimespan(
1126         start_offset=start_offset,
1127         stop_offset=stop_offset,
1128         annotation=MusicSpecifier(
1129             music_maker=music_maker,
1130             voice_name='Voice 9',
1131         ),
1132     )
1133     for start_offset, stop_offset, music_maker in [
1134         [(0, 8), (2, 8), alto_five_musicmaker_two],
1135         [(2, 8), (4, 8), alto_five_musicmaker_one],
1136         [(4, 8), (6, 8), alto_five_musicmaker_one],
1137         [(6, 8), (8, 8), alto_five_musicmaker_two],
1138         [(8, 8), (10, 8), alto_five_musicmaker_three],
1139
1140         [(12, 8), (14, 8), alto_five_musicmaker_one],
1141         [(14, 8), (16, 8), alto_five_musicmaker_one],
1142         [(16, 8), (18, 8), alto_five_musicmaker_one],
1143         [(18, 8), (20, 8), alto_five_musicmaker_one],
1144         [(20, 8), (22, 8), alto_five_musicmaker_three],
1145         [(22, 8), (24, 8), alto_five_musicmaker_two],
1146         [(24, 8), (26, 8), alto_five_musicmaker_two],
1147
1148         [(28, 8), (30, 8), alto_five_musicmaker_two],
1149         [(30, 8), (32, 8), alto_five_musicmaker_three],
1150         [(32, 8), (34, 8), alto_five_musicmaker_one],
1151         [(34, 8), (36, 8), alto_five_musicmaker_one],
1152         [(36, 8), (38, 8), alto_five_musicmaker_one],
1153         [(38, 8), (40, 8), alto_five_musicmaker_three],
1154         [(40, 8), (42, 8), alto_five_musicmaker_one],
1155         [(42, 8), (44, 8), alto_five_musicmaker_two],
1156         [(44, 8), (46, 8), alto_five_musicmaker_two],
1157
1158         [(48, 8), (50, 8), alto_five_musicmaker_three],
1159         [(50, 8), (52, 8), alto_five_musicmaker_two],
1160         [(52, 8), (54, 8), alto_five_musicmaker_two],
1161         [(54, 8), (56, 8), alto_five_musicmaker_two],
1162     ]
1163 ])
1164
1165 voice_10_timespan_list = abjad.TimespanList([
1166     abjad.AnnotatedTimespan(
1167         start_offset=start_offset,
1168         stop_offset=stop_offset,
1169         annotation=MusicSpecifier(
1170             music_maker=music_maker,
1171             voice_name='Voice 10',
1172         ),
1173     )
1174     for start_offset, stop_offset, music_maker in [
1175         [(0, 8), (2, 8), alto_six_musicmaker_one],

```

```

1176     [(4, 8), (6, 8), alto_six_musicmaker_three],
1177     [(6, 8), (8, 8), alto_six_musicmaker_one],
1178     [(8, 8), (10, 8), alto_six_musicmaker_two],
1179     [(10, 8), (12, 8), alto_six_musicmaker_three],
1180     [(12, 8), (14, 8), alto_six_musicmaker_two],
1181     [(14, 8), (16, 8), alto_six_musicmaker_two],
1182     [(16, 8), (18, 8), alto_six_musicmaker_one],
1183
1184     [(20, 8), (22, 8), alto_six_musicmaker_one],
1185     [(22, 8), (24, 8), alto_six_musicmaker_one],
1186     [(24, 8), (26, 8), alto_six_musicmaker_one],
1187     [(26, 8), (28, 8), alto_six_musicmaker_three],
1188     [(28, 8), (30, 8), alto_six_musicmaker_two],
1189     [(30, 8), (32, 8), alto_six_musicmaker_two],
1190     [(32, 8), (34, 8), alto_six_musicmaker_two],
1191     [(34, 8), (36, 8), alto_six_musicmaker_two],
1192     [(36, 8), (38, 8), alto_six_musicmaker_three],
1193     [(38, 8), (40, 8), alto_six_musicmaker_one],
1194     [(40, 8), (42, 8), alto_six_musicmaker_one],
1195     [(42, 8), (44, 8), alto_six_musicmaker_one],
1196     [(44, 8), (46, 8), alto_six_musicmaker_one],
1197     [(46, 8), (48, 8), alto_six_musicmaker_three],
1198     [(48, 8), (50, 8), alto_six_musicmaker_one],
1199
1200     [(52, 8), (54, 8), alto_six_musicmaker_two],
1201     [(54, 8), (56, 8), alto_six_musicmaker_two],
1202 ]
1203 ])
1204
1205
1206 voice_11_timespan_list = abjad.TimespanList([
1207     abjad.AnnotatedTimespan(
1208         start_offset=start_offset,
1209         stop_offset=stop_offset,
1210         annotation=MusicSpecifier(
1211             music_maker=music_maker,
1212             voice_name='Voice 11',
1213         ),
1214     )
1215     for start_offset, stop_offset, music_maker in [
1216         [(0, 8), (2, 8), tenor_one_musicmaker_one],
1217         [(2, 8), (4, 8), tenor_one_musicmaker_one],
1218         [(4, 8), (6, 8), tenor_one_musicmaker_two],
1219         [(6, 8), (8, 8), tenor_one_musicmaker_two],
1220         [(8, 8), (10, 8), tenor_one_musicmaker_three],
1221         [(10, 8), (12, 8), tenor_one_musicmaker_one],
1222         [(12, 8), (14, 8), tenor_one_musicmaker_one],
1223
1224         [(16, 8), (18, 8), tenor_one_musicmaker_one],
1225         [(18, 8), (20, 8), tenor_one_musicmaker_two],
1226         [(20, 8), (22, 8), tenor_one_musicmaker_two],
1227         [(22, 8), (24, 8), tenor_one_musicmaker_three],
1228         [(24, 8), (26, 8), tenor_one_musicmaker_two],
1229         [(26, 8), (28, 8), tenor_one_musicmaker_one],
1230         [(28, 8), (30, 8), tenor_one_musicmaker_one],
1231         [(30, 8), (32, 8), tenor_one_musicmaker_one],
1232         [(32, 8), (34, 8), tenor_one_musicmaker_one],
1233         [(34, 8), (36, 8), tenor_one_musicmaker_one],
1234         [(36, 8), (38, 8), tenor_one_musicmaker_two],

```

```

1235     [(40, 8), (42, 8), tenor_one_musicmaker_three],
1236     [(42, 8), (44, 8), tenor_one_musicmaker_two],
1237     [(44, 8), (46, 8), tenor_one_musicmaker_one],
1238     [(46, 8), (48, 8), tenor_one_musicmaker_one],
1239     [(48, 8), (50, 8), tenor_one_musicmaker_one],
1240     [(50, 8), (52, 8), tenor_one_musicmaker_one],
1241     [(52, 8), (54, 8), tenor_one_musicmaker_two],
1242     [(54, 8), (56, 8), tenor_one_musicmaker_two],
1243   ],
1244 ]
1245 ])
1246
1247 voice_12_timespan_list = abjad.TimespanList([
1248     abjad.AnnotatedTimespan(
1249         start_offset=start_offset,
1250         stop_offset=stop_offset,
1251         annotation=MusicSpecifier(
1252             music_maker=music_maker,
1253             voice_name='Voice 12',
1254         ),
1255     ),
1256     for start_offset, stop_offset, music_maker in [
1257         [(0, 8), (2, 8), tenor_two_musicmaker_two],
1258         [(2, 8), (4, 8), tenor_two_musicmaker_one],
1259         [(4, 8), (6, 8), tenor_two_musicmaker_one],
1260         [(6, 8), (8, 8), tenor_two_musicmaker_one],
1261         [(8, 8), (10, 8), tenor_two_musicmaker_two],
1262         [(10, 8), (12, 8), tenor_two_musicmaker_one],
1263         [(12, 8), (14, 8), tenor_two_musicmaker_one],
1264         [(14, 8), (16, 8), tenor_two_musicmaker_two],
1265         [(16, 8), (18, 8), tenor_two_musicmaker_three],
1266         [(18, 8), (20, 8), tenor_two_musicmaker_two],
1267
1268         [(22, 8), (24, 8), tenor_two_musicmaker_two],
1269         [(24, 8), (26, 8), tenor_two_musicmaker_two],
1270         [(26, 8), (28, 8), tenor_two_musicmaker_two],
1271         [(28, 8), (30, 8), tenor_two_musicmaker_one],
1272         [(30, 8), (32, 8), tenor_two_musicmaker_one],
1273         [(32, 8), (34, 8), tenor_two_musicmaker_three],
1274         [(34, 8), (36, 8), tenor_two_musicmaker_one],
1275         [(36, 8), (38, 8), tenor_two_musicmaker_one],
1276         [(38, 8), (39, 8), tenor_two_musicmaker_one],
1277         [(40, 8), (42, 8), tenor_two_musicmaker_one],
1278         [(42, 8), (44, 8), tenor_two_musicmaker_two],
1279
1280         [(46, 8), (48, 8), tenor_two_musicmaker_two],
1281         [(48, 8), (50, 8), tenor_two_musicmaker_one],
1282         [(51, 8), (52, 8), tenor_two_musicmaker_one],
1283         [(52, 8), (54, 8), tenor_two_musicmaker_three],
1284         [(54, 8), (56, 8), tenor_two_musicmaker_two],
1285     ],
1286   ])
1287
1288 voice_13_timespan_list = abjad.TimespanList([
1289     abjad.AnnotatedTimespan(
1290         start_offset=start_offset,
1291         stop_offset=stop_offset,
1292         annotation=MusicSpecifier(
1293             music_maker=music_maker,

```

```

1294     voice_name='Voice 13',
1295     ),
1296 )
1297 for start_offset, stop_offset, music_maker in [
1298     [(0, 8), (2, 8), tenor_three_musicmaker_three],
1299     [(2, 8), (4, 8), tenor_three_musicmaker_one],
1300     [(4, 8), (6, 8), tenor_three_musicmaker_three],
1301     [(6, 8), (8, 8), tenor_three_musicmaker_one],
1302     [(8, 8), (10, 8), tenor_three_musicmaker_one],
1303     [(10, 8), (12, 8), tenor_three_musicmaker_three],
1304     [(12, 8), (14, 8), tenor_three_musicmaker_two],
1305     [(14, 8), (16, 8), tenor_three_musicmaker_two],
1306
1307     [(18, 8), (20, 8), tenor_three_musicmaker_three],
1308     [(20, 8), (22, 8), tenor_three_musicmaker_two],
1309     [(22, 8), (24, 8), tenor_three_musicmaker_two],
1310     [(24, 8), (26, 8), tenor_three_musicmaker_one],
1311     [(26, 8), (28, 8), tenor_three_musicmaker_one],
1312     [(28, 8), (30, 8), tenor_three_musicmaker_three],
1313     [(30, 8), (32, 8), tenor_three_musicmaker_two],
1314     [(32, 8), (34, 8), tenor_three_musicmaker_two],
1315     [(34, 8), (36, 8), tenor_three_musicmaker_two],
1316
1317     [(38, 8), (40, 8), tenor_three_musicmaker_one],
1318     [(40, 8), (42, 8), tenor_three_musicmaker_three],
1319     [(42, 8), (44, 8), tenor_three_musicmaker_one],
1320     [(44, 8), (46, 8), tenor_three_musicmaker_one],
1321     [(46, 8), (48, 8), tenor_three_musicmaker_one],
1322
1323     [(50, 8), (52, 8), tenor_three_musicmaker_three],
1324     [(52, 8), (54, 8), tenor_three_musicmaker_two],
1325     [(54, 8), (56, 8), tenor_three_musicmaker_two],
1326 ]
1327 ])
1328
1329 voice_14_timespan_list = abjad.TimespanList([
1330     abjad.AnnotatedTimespan(
1331         start_offset=start_offset,
1332         stop_offset=stop_offset,
1333         annotation=MusicSpecifier(
1334             music_maker=music_maker,
1335             voice_name='Voice 14',
1336         ),
1337     )
1338     for start_offset, stop_offset, music_maker in [
1339         [(0, 8), (2, 8), tenor_four_musicmaker_two],
1340         [(2, 8), (4, 8), tenor_four_musicmaker_two],
1341         [(4, 8), (6, 8), tenor_four_musicmaker_two],
1342         [(6, 8), (8, 8), tenor_four_musicmaker_two],
1343         [(8, 8), (10, 8), tenor_four_musicmaker_two],
1344         [(10, 8), (12, 8), tenor_four_musicmaker_one],
1345         [(12, 8), (14, 8), tenor_four_musicmaker_two],
1346         [(14, 8), (16, 8), tenor_four_musicmaker_two],
1347         [(16, 8), (18, 8), tenor_four_musicmaker_one],
1348
1349         [(22, 8), (24, 8), tenor_four_musicmaker_one],
1350         [(24, 8), (26, 8), tenor_four_musicmaker_three],
1351         [(26, 8), (28, 8), tenor_four_musicmaker_one],
1352         [(28, 8), (30, 8), tenor_four_musicmaker_one],

```

```

1353     [(30, 8), (32, 8), tenor_four_musicmaker_one],
1354     [(32, 8), (34, 8), tenor_four_musicmaker_two],
1355     [(34, 8), (36, 8), tenor_four_musicmaker_two],
1356     [(36, 8), (38, 8), tenor_four_musicmaker_two],
1357     [(38, 8), (40, 8), tenor_four_musicmaker_three],
1358     [(40, 8), (42, 8), tenor_four_musicmaker_two],
1359
1360     [(46, 8), (48, 8), tenor_four_musicmaker_two],
1361     [(48, 8), (50, 8), tenor_four_musicmaker_one],
1362     [(50, 8), (52, 8), tenor_four_musicmaker_three],
1363     [(52, 8), (54, 8), tenor_four_musicmaker_one],
1364     [(54, 8), (56, 8), tenor_four_musicmaker_one],
1365 ]
1366 ])
1367
1368 voice_15_timespan_list = abjad.TimespanList([
1369     abjad.AnnotatedTimespan(
1370         start_offset=start_offset,
1371         stop_offset=stop_offset,
1372         annotation=MusicSpecifier(
1373             music_maker=music_maker,
1374             voice_name='Voice 15',
1375         ),
1376     )
1377     for start_offset, stop_offset, music_maker in [
1378         [(0, 8), (2, 8), tenor_five_musicmaker_two],
1379         [(2, 8), (4, 8), tenor_five_musicmaker_two],
1380         [(4, 8), (6, 8), tenor_five_musicmaker_two],
1381         [(6, 8), (8, 8), tenor_five_musicmaker_one],
1382         [(8, 8), (10, 8), tenor_five_musicmaker_three],
1383         [(10, 8), (12, 8), tenor_five_musicmaker_one],
1384
1385         [(14, 8), (16, 8), tenor_five_musicmaker_two],
1386         [(16, 8), (18, 8), tenor_five_musicmaker_two],
1387         [(18, 8), (20, 8), tenor_five_musicmaker_three],
1388         [(20, 8), (22, 8), tenor_five_musicmaker_one],
1389         [(22, 8), (24, 8), tenor_five_musicmaker_one],
1390         [(24, 8), (26, 8), tenor_five_musicmaker_one],
1391         [(26, 8), (28, 8), tenor_five_musicmaker_one],
1392         [(28, 8), (30, 8), tenor_five_musicmaker_three],
1393
1394         [(32, 8), (34, 8), tenor_five_musicmaker_two],
1395         [(34, 8), (36, 8), tenor_five_musicmaker_one],
1396         [(36, 8), (38, 8), tenor_five_musicmaker_one],
1397         [(38, 8), (40, 8), tenor_five_musicmaker_three],
1398         [(40, 8), (42, 8), tenor_five_musicmaker_one],
1399         [(42, 8), (44, 8), tenor_five_musicmaker_two],
1400         [(44, 8), (46, 8), tenor_five_musicmaker_two],
1401
1402         [(50, 8), (52, 8), tenor_five_musicmaker_one],
1403         [(52, 8), (54, 8), tenor_five_musicmaker_one],
1404         [(54, 8), (56, 8), tenor_five_musicmaker_one],
1405     ]
1406 ])
1407
1408 voice_16_timespan_list = abjad.TimespanList([
1409     abjad.AnnotatedTimespan(
1410         start_offset=start_offset,
1411         stop_offset=stop_offset,

```

```

1412     annotation=MusicSpecifier(
1413         music_maker=music_maker,
1414         voice_name='Voice 16',
1415     ),
1416 )
1417 for start_offset, stop_offset, music_maker in [
1418     [(0, 8), (2, 8), baritone_one_musicmaker_two],
1419     [(2, 8), (4, 8), baritone_one_musicmaker_two],
1420     [(4, 8), (6, 8), baritone_one_musicmaker_two],
1421
1422     [(8, 8), (10, 8), baritone_one_musicmaker_one],
1423     [(10, 8), (12, 8), baritone_one_musicmaker_three],
1424     [(12, 8), (14, 8), baritone_one_musicmaker_one],
1425     [(14, 8), (16, 8), baritone_one_musicmaker_two],
1426
1427     [(18, 8), (20, 8), baritone_one_musicmaker_two],
1428     [(20, 8), (22, 8), baritone_one_musicmaker_two],
1429     [(22, 8), (24, 8), baritone_one_musicmaker_one],
1430     [(24, 8), (26, 8), baritone_one_musicmaker_one],
1431
1432     [(28, 8), (30, 8), baritone_one_musicmaker_two],
1433     [(30, 8), (32, 8), baritone_one_musicmaker_two],
1434     [(32, 8), (34, 8), baritone_one_musicmaker_two],
1435     [(34, 8), (36, 8), baritone_one_musicmaker_two],
1436     [(36, 8), (38, 8), baritone_one_musicmaker_one],
1437     [(38, 8), (40, 8), baritone_one_musicmaker_three],
1438     [(40, 8), (42, 8), baritone_one_musicmaker_one],
1439     [(42, 8), (44, 8), baritone_one_musicmaker_two],
1440
1441     [(46, 8), (48, 8), baritone_one_musicmaker_two],
1442     [(48, 8), (50, 8), baritone_one_musicmaker_two],
1443     [(50, 8), (52, 8), baritone_one_musicmaker_one],
1444     [(52, 8), (54, 8), baritone_one_musicmaker_one],
1445     [(54, 8), (56, 8), baritone_one_musicmaker_three],
1446 ]
1447 ])
1448
1449 voice_17_timespan_list = abjad.TimespanList([
1450     abjad.AnnotatedTimespan(
1451         start_offset=start_offset,
1452         stop_offset=stop_offset,
1453         annotation=MusicSpecifier(
1454             music_maker=music_maker,
1455             voice_name='Voice 17',
1456         ),
1457     )
1458     for start_offset, stop_offset, music_maker in [
1459         [(0, 8), (2, 8), baritone_two_musicmaker_one],
1460         [(2, 8), (4, 8), baritone_two_musicmaker_one],
1461         [(4, 8), (6, 8), baritone_two_musicmaker_two],
1462         [(6, 8), (8, 8), baritone_two_musicmaker_two],
1463         [(8, 8), (10, 8), baritone_two_musicmaker_two],
1464         [(10, 8), (12, 8), baritone_two_musicmaker_three],
1465         [(12, 8), (14, 8), baritone_two_musicmaker_one],
1466         [(14, 8), (16, 8), baritone_two_musicmaker_one],
1467         [(16, 8), (18, 8), baritone_two_musicmaker_one],
1468
1469         [(22, 8), (24, 8), baritone_two_musicmaker_two],
1470         [(24, 8), (26, 8), baritone_two_musicmaker_two],

```

```

1471     [(26, 8), (28, 8), baritone_two_musicmaker_two],
1472     [(28, 8), (30, 8), baritone_two_musicmaker_three],
1473     [(30, 8), (32, 8), baritone_two_musicmaker_one],
1474     [(32, 8), (34, 8), baritone_two_musicmaker_one],
1475
1476     [(36, 8), (38, 8), baritone_two_musicmaker_one],
1477     [(38, 8), (40, 8), baritone_two_musicmaker_one],
1478     [(40, 8), (42, 8), baritone_two_musicmaker_two],
1479     [(42, 8), (44, 8), baritone_two_musicmaker_two],
1480     [(44, 8), (46, 8), baritone_two_musicmaker_two],
1481     [(46, 8), (48, 8), baritone_two_musicmaker_one],
1482     [(48, 8), (50, 8), baritone_two_musicmaker_two],
1483     [(50, 8), (52, 8), baritone_two_musicmaker_two],
1484     [(52, 8), (54, 8), baritone_two_musicmaker_two],
1485     [(54, 8), (56, 8), baritone_two_musicmaker_three],
1486 ]
1487 ])
1488
1489 voice_18_timespan_list = abjad.TimespanList([
1490     abjad.AnnotatedTimespan(
1491         start_offset=start_offset,
1492         stop_offset=stop_offset,
1493         annotation=MusicSpecifier(
1494             music_maker=music_maker,
1495             voice_name='Voice 18',
1496         ),
1497     )
1498     for start_offset, stop_offset, music_maker in [
1499         [(0, 8), (2, 8), baritone_three_musicmaker_one],
1500         [(2, 8), (4, 8), baritone_three_musicmaker_one],
1501         [(4, 8), (6, 8), baritone_three_musicmaker_one],
1502         [(6, 8), (8, 8), baritone_three_musicmaker_one],
1503         [(8, 8), (10, 8), baritone_three_musicmaker_three],
1504         [(10, 8), (12, 8), baritone_three_musicmaker_two],
1505
1506         [(16, 8), (18, 8), baritone_three_musicmaker_one],
1507
1508         [(20, 8), (22, 8), baritone_three_musicmaker_one],
1509         [(22, 8), (24, 8), baritone_three_musicmaker_one],
1510         [(24, 8), (26, 8), baritone_three_musicmaker_three],
1511         [(26, 8), (28, 8), baritone_three_musicmaker_two],
1512         [(28, 8), (30, 8), baritone_three_musicmaker_two],
1513         [(30, 8), (32, 8), baritone_three_musicmaker_two],
1514
1515         [(36, 8), (38, 8), baritone_three_musicmaker_one],
1516         [(39, 8), (40, 8), baritone_three_musicmaker_one],
1517         [(40, 8), (42, 8), baritone_three_musicmaker_three],
1518         [(42, 8), (44, 8), baritone_three_musicmaker_two],
1519         [(44, 8), (46, 8), baritone_three_musicmaker_two],
1520         [(46, 8), (48, 8), baritone_three_musicmaker_two],
1521
1522         [(50, 8), (51, 8), baritone_three_musicmaker_one],
1523         [(52, 8), (54, 8), baritone_three_musicmaker_one],
1524         [(54, 8), (56, 8), baritone_three_musicmaker_one],
1525     ]
1526 ])
1527
1528 voice_19_timespan_list = abjad.TimespanList([
1529     abjad.AnnotatedTimespan(

```

```

1530     start_offset=start_offset,
1531     stop_offset=stop_offset,
1532     annotation=MusicSpecifier(
1533         music_maker=music_maker,
1534         voice_name='Voice 19',
1535     ),
1536 )
1537 for start_offset, stop_offset, music_maker in [
1538     [(0, 8), (2, 8), bass_one_musicmaker_two],
1539     [(2, 8), (4, 8), bass_one_musicmaker_two],
1540     [(4, 8), (6, 8), bass_one_musicmaker_two],
1541     [(6, 8), (8, 8), bass_one_musicmaker_one],
1542     [(8, 8), (10, 8), bass_one_musicmaker_one],
1543     [(10, 8), (12, 8), bass_one_musicmaker_two],
1544     [(12, 8), (14, 8), bass_one_musicmaker_two],
1545     [(14, 8), (16, 8), bass_one_musicmaker_two],
1546     [(16, 8), (18, 8), bass_one_musicmaker_three],
1547     [(18, 8), (20, 8), bass_one_musicmaker_one],
1548
1549     [(22, 8), (24, 8), bass_one_musicmaker_two],
1550     [(24, 8), (26, 8), bass_one_musicmaker_two],
1551     [(26, 8), (28, 8), bass_one_musicmaker_two],
1552     [(28, 8), (30, 8), bass_one_musicmaker_three],
1553     [(30, 8), (32, 8), bass_one_musicmaker_one],
1554     [(32, 8), (34, 8), bass_one_musicmaker_one],
1555     [(34, 8), (36, 8), bass_one_musicmaker_three],
1556
1557     [(38, 8), (40, 8), bass_one_musicmaker_one],
1558     [(40, 8), (42, 8), bass_one_musicmaker_two],
1559     [(42, 8), (44, 8), bass_one_musicmaker_two],
1560     [(44, 8), (46, 8), bass_one_musicmaker_one],
1561     [(46, 8), (48, 8), bass_one_musicmaker_two],
1562     [(48, 8), (50, 8), bass_one_musicmaker_two],
1563     [(50, 8), (52, 8), bass_one_musicmaker_two],
1564     [(52, 8), (54, 8), bass_one_musicmaker_three],
1565 ]
1566 ])
1567
1568 voice_20_timespan_list = abjad.TimespanList([
1569     abjad.AnnotatedTimespan(
1570         start_offset=start_offset,
1571         stop_offset=stop_offset,
1572         annotation=MusicSpecifier(
1573             music_maker=music_maker,
1574             voice_name='Voice 20',
1575         ),
1576     )
1577 for start_offset, stop_offset, music_maker in [
1578     [(0, 8), (2, 8), bass_two_musicmaker_two],
1579     [(2, 8), (4, 8), bass_two_musicmaker_two],
1580     [(4, 8), (6, 8), bass_two_musicmaker_two],
1581     [(6, 8), (8, 8), bass_two_musicmaker_one],
1582     [(8, 8), (10, 8), bass_two_musicmaker_one],
1583     [(10, 8), (12, 8), bass_two_musicmaker_two],
1584     [(12, 8), (14, 8), bass_two_musicmaker_two],
1585
1586     [(16, 8), (18, 8), bass_two_musicmaker_two],
1587     [(18, 8), (20, 8), bass_two_musicmaker_one],
1588     [(20, 8), (22, 8), bass_two_musicmaker_three],

```

```

1589     [(22, 8), (24, 8), bass_two_musicmaker_two],
1590     [(24, 8), (26, 8), bass_two_musicmaker_two],
1591     [(26, 8), (28, 8), bass_two_musicmaker_two],
1592     [(28, 8), (30, 8), bass_two_musicmaker_two],
1593     [(30, 8), (32, 8), bass_two_musicmaker_three],
1594     [(32, 8), (34, 8), bass_two_musicmaker_one],
1595
1596     [(36, 8), (38, 8), bass_two_musicmaker_two],
1597     [(38, 8), (40, 8), bass_two_musicmaker_one],
1598     [(40, 8), (42, 8), bass_two_musicmaker_one],
1599     [(42, 8), (44, 8), bass_two_musicmaker_one],
1600     [(44, 8), (46, 8), bass_two_musicmaker_one],
1601     [(46, 8), (48, 8), bass_two_musicmaker_two],
1602     [(48, 8), (50, 8), bass_two_musicmaker_three],
1603
1604     [(52, 8), (54, 8), bass_two_musicmaker_two],
1605     [(54, 8), (56, 8), bass_two_musicmaker_two],
1606 ]
1607 ])
1608
1609 voice_21_timestrap_list = abjad.TimespanList([
1610     abjad.AnnotatedTimespan(
1611         start_offset=start_offset,
1612         stop_offset=stop_offset,
1613         annotation=MusicSpecifier(
1614             music_maker=music_maker,
1615             voice_name='Voice 21',
1616         ),
1617     )
1618     for start_offset, stop_offset, music_maker in [
1619         [(0, 8), (2, 8), contrabass_musicmaker_two],
1620         [(2, 8), (4, 8), contrabass_musicmaker_two],
1621         [(4, 8), (6, 8), contrabass_musicmaker_two],
1622         [(6, 8), (8, 8), contrabass_musicmaker_three],
1623         [(8, 8), (10, 8), contrabass_musicmaker_one],
1624         [(10, 8), (12, 8), contrabass_musicmaker_two],
1625
1626         [(14, 8), (16, 8), contrabass_musicmaker_two],
1627         [(16, 8), (18, 8), contrabass_musicmaker_two],
1628         [(18, 8), (20, 8), contrabass_musicmaker_three],
1629         [(20, 8), (22, 8), contrabass_musicmaker_two],
1630         [(22, 8), (24, 8), contrabass_musicmaker_two],
1631         [(24, 8), (26, 8), contrabass_musicmaker_two],
1632
1633         [(30, 8), (32, 8), contrabass_musicmaker_one],
1634         [(32, 8), (34, 8), contrabass_musicmaker_one],
1635         [(34, 8), (36, 8), contrabass_musicmaker_one],
1636         [(36, 8), (38, 8), contrabass_musicmaker_one],
1637         [(38, 8), (40, 8), contrabass_musicmaker_three],
1638         [(40, 8), (42, 8), contrabass_musicmaker_one],
1639         [(42, 8), (44, 8), contrabass_musicmaker_one],
1640         [(44, 8), (46, 8), contrabass_musicmaker_one],
1641         [(46, 8), (48, 8), contrabass_musicmaker_three],
1642         [(48, 8), (50, 8), contrabass_musicmaker_two],
1643         [(50, 8), (52, 8), contrabass_musicmaker_two],
1644         [(52, 8), (54, 8), contrabass_musicmaker_two],
1645         [(54, 8), (56, 8), contrabass_musicmaker_two],
1646     ]
1647 ])

```

```

1648 all_timespan_lists = {
1649     'Voice 1': voice_1_timespan_list,
1650     'Voice 2': voice_2_timespan_list,
1651     'Voice 3': voice_3_timespan_list,
1652     'Voice 4': voice_4_timespan_list,
1653     'Voice 5': voice_5_timespan_list,
1654     'Voice 6': voice_6_timespan_list,
1655     'Voice 7': voice_7_timespan_list,
1656     'Voice 8': voice_8_timespan_list,
1657     'Voice 9': voice_9_timespan_list,
1658     'Voice 10': voice_10_timespan_list,
1659     'Voice 11': voice_11_timespan_list,
1660     'Voice 12': voice_12_timespan_list,
1661     'Voice 13': voice_13_timespan_list,
1662     'Voice 14': voice_14_timespan_list,
1663     'Voice 15': voice_15_timespan_list,
1664     'Voice 16': voice_16_timespan_list,
1665     'Voice 17': voice_17_timespan_list,
1666     'Voice 18': voice_18_timespan_list,
1667     'Voice 19': voice_19_timespan_list,
1668     'Voice 20': voice_20_timespan_list,
1669     'Voice 21': voice_21_timespan_list,
1670 }
1671 }
1672
1673 global_timespan = abjad.Timespan(
1674     start_offset=0,
1675     stop_offset=max(_.stop_offset for _ in all_timespan_lists.values())
1676 )
1677
1678 for voice_name, timespan_list in all_timespan_lists.items():
1679     silences = abjad.TimespanList([global_timespan])
1680     silences.extend(timespan_list)
1681     silences.sort()
1682     silences.compute_logical_xor()
1683     for silence_timespan in silences:
1684         timespan_list.append(
1685             abjad.AnnotatedTimespan(
1686                 start_offset=silence_timespan.start_offset,
1687                 stop_offset=silence_timespan.stop_offset,
1688                 annotation=MusicSpecifier(
1689                     music_maker=None,
1690                     voice_name=voice_name,
1691                 ),
1692             ),
1693         )
1694     timespan_list.sort()
1695
1696 for voice_name, timespan_list in all_timespan_lists.items():
1697     shards = timespan_list.split_at_offsets(bounds)
1698     split_timespan_list = abjad.TimespanList()
1699     for shard in shards:
1700         split_timespan_list.extend(shard)
1701     split_timespan_list.sort()
1702     all_timespan_lists[voice_name] = timespan_list
1703
1704 score = abjad.Score([
1705     abjad.Staff(lilypond_type='TimeSignatureContext', name='Global Context'),
1706     abjad.StaffGroup(

```

```

1707     [
1708         abjad.Staff([abjad.Voice(name='Voice 1')], name='Staff 1',
1709                     lilypond_type='Staff',),
1710                     abjad.Staff([abjad.Voice(name='Voice 2')], name='Staff 2',
1711                     lilypond_type='Staff',),
1712                     abjad.Staff([abjad.Voice(name='Voice 3')], name='Staff 3',
1713                     lilypond_type='Staff',),
1714                     abjad.Staff([abjad.Voice(name='Voice 4')], name='Staff 4',
1715                     lilypond_type='Staff',),
1716                     abjad.Staff([abjad.Voice(name='Voice 5')], name='Staff 5',
1717                     lilypond_type='Staff',),
1718                     abjad.Staff([abjad.Voice(name='Voice 6')], name='Staff 6',
1719                     lilypond_type='Staff',),
1720                     abjad.Staff([abjad.Voice(name='Voice 7')], name='Staff 7',
1721                     lilypond_type='Staff',),
1722                     abjad.Staff([abjad.Voice(name='Voice 8')], name='Staff 8',
1723                     lilypond_type='Staff',),
1724                     abjad.Staff([abjad.Voice(name='Voice 9')], name='Staff 9',
1725                     lilypond_type='Staff',),
1726                     abjad.Staff([abjad.Voice(name='Voice 10')], name='Staff 10',
1727                     lilypond_type='Staff',),
1728                     abjad.Staff([abjad.Voice(name='Voice 11')], name='Staff 11',
1729                     lilypond_type='Staff',),
1730                     abjad.Staff([abjad.Voice(name='Voice 12')], name='Staff 12',
1731                     lilypond_type='Staff',),
1732                     abjad.Staff([abjad.Voice(name='Voice 13')], name='Staff 13',
1733                     lilypond_type='Staff',),
1734                     abjad.Staff([abjad.Voice(name='Voice 14')], name='Staff 14',
1735                     lilypond_type='Staff',),
1736                     abjad.Staff([abjad.Voice(name='Voice 15')], name='Staff 15',
1737                     lilypond_type='Staff',),
1738                     abjad.Staff([abjad.Voice(name='Voice 16')], name='Staff 16',
1739                     lilypond_type='Staff',),
1740                     abjad.Staff([abjad.Voice(name='Voice 17')], name='Staff 17',
1741                     lilypond_type='Staff',),
1742                     abjad.Staff([abjad.Voice(name='Voice 18')], name='Staff 18',
1743                     lilypond_type='Staff',),
1744                     abjad.Staff([abjad.Voice(name='Voice 19')], name='Staff 19',
1745                     lilypond_type='Staff',),
1746                     abjad.Staff([abjad.Voice(name='Voice 20')], name='Staff 20',
1747                     lilypond_type='Staff',),
1748                     abjad.Staff([abjad.Voice(name='Voice 21')], name='Staff 21',
1749                     lilypond_type='Staff',),
1750                     ],
1751                     name='Staff Group',
1752                 )
1753             ])
1754
1755         for time_signature in time_signatures:
1756             skip = abjad.Skip(1, multiplier=(time_signature))
1757             abjad.attach(time_signature, skip)
1758             score['Global Context'].append(skip)
1759
1760         print('Making containers ...')
1761
1762     def make_container(music_maker, durations):
1763         selections = music_maker(durations)
1764         container = abjad.Container([])
1765         container.extend(selections)

```

```

1745     return container
1746
1747 def key_function(timespan):
1748     return timespan.annotation.music_maker or silence_maker
1749
1750 for voice_name, timespan_list in all_timespan_lists.items():
1751     for music_maker, grouper in itertools.groupby(
1752         timespan_list,
1753         key=key_function,
1754     ):
1755         durations = [timespan.duration for timespan in grouper]
1756         container = make_container(music_maker, durations)
1757         voice = score[voice_name]
1758         voice.append(container)
1759
1760 print('Splitting and rewriting ...')
1761 for voice in abjad.iterate(score['Staff Group']).components(abjad.Voice):
1762     for i, shard in enumerate(abjad.mutate(voice[:]).split(time_signatures)):
1763         time_signature = time_signatures[i]
1764         abjad.mutate(shard).rewrite_meter(time_signature)
1765
1766 print('Beaming runs ...')
1767 for voice in abjad.select(score).components(abjad.Voice):
1768     for run in abjad.select(voice).runs():
1769         specifier = abjadext.rmakers.BeamSpecifier(
1770             beam_each_division=False,
1771             )
1772         specifier(run)
1773     abjad.beam(voice[:, beam_lone_notes=False, beam_rests=False])
1774
1775 print('Stopping Hairpins ...')
1776 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1777     for rest in abjad.iterate(staff).components(abjad.Rest):
1778         previous_leaf = abjad.inspect(rest).leaf(-1)
1779         if isinstance(previous_leaf, abjad.Note):
1780             abjad.attach(abjad.StopHairpin(), rest)
1781         elif isinstance(previous_leaf, abjad.Chord):
1782             abjad.attach(abjad.StopHairpin(), rest)
1783         elif isinstance(previous_leaf, abjad.Rest):
1784             pass
1785
1786 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1787     first_leaf = abjad.select(staff).leaves()[0]
1788     stop = abjad.LilyPondLiteral(r'\!', format_slot='after',)
1789     abjad.attach(stop, first_leaf)
1790
1791 print('Adding attachments ...')
1792 bar_line = abjad.BarLine('|.')
1793 markup = abjad.Markup(r'\bold { J }')
1794 mark = abjad.RehearsalMark(markup=markup)
1795
1796 instruments = cyc([
1797     abjad.SopranoSaxophone(),
1798     abjad.SopranoSaxophone(),
1799     abjad.SopranoSaxophone(),
1800     abjad.SopranoSaxophone(),
1801     abjad.AltoSaxophone(),
1802     abjad.AltoSaxophone(),
1803     abjad.AltoSaxophone(),

```

```

1804     abjad.AltoSaxophone(),
1805     abjad.AltoSaxophone(),
1806     abjad.AltoSaxophone(),
1807     abjad.TenorSaxophone(),
1808     abjad.TenorSaxophone(),
1809     abjad.TenorSaxophone(),
1810     abjad.TenorSaxophone(),
1811     abjad.TenorSaxophone(),
1812     abjad.BaritoneSaxophone(),
1813     abjad.BaritoneSaxophone(),
1814     abjad.BaritoneSaxophone(),
1815     abjad.BassSaxophone(),
1816     abjad.BassSaxophone(),
1817     abjad.ContrabassSaxophone(),
1818 ])
1819
1820 abbreviations = cyc([
1821     abjad.MarginMarkup(markup=abjad.Markup('spro.'),),
1822     abjad.MarginMarkup(markup=abjad.Markup('spr.1'),),
1823     abjad.MarginMarkup(markup=abjad.Markup('spr.2'),),
1824     abjad.MarginMarkup(markup=abjad.Markup('spr.3'),),
1825     abjad.MarginMarkup(markup=abjad.Markup('alt.1'),),
1826     abjad.MarginMarkup(markup=abjad.Markup('alt.2'),),
1827     abjad.MarginMarkup(markup=abjad.Markup('alt.3'),),
1828     abjad.MarginMarkup(markup=abjad.Markup('alt.4'),),
1829     abjad.MarginMarkup(markup=abjad.Markup('alt.5'),),
1830     abjad.MarginMarkup(markup=abjad.Markup('alt.6'),),
1831     abjad.MarginMarkup(markup=abjad.Markup('ten.1'),),
1832     abjad.MarginMarkup(markup=abjad.Markup('ten.2'),),
1833     abjad.MarginMarkup(markup=abjad.Markup('ten.3'),),
1834     abjad.MarginMarkup(markup=abjad.Markup('ten.4'),),
1835     abjad.MarginMarkup(markup=abjad.Markup('ten.5'),),
1836     abjad.MarginMarkup(markup=abjad.Markup('bar.1'),),
1837     abjad.MarginMarkup(markup=abjad.Markup('bar.2'),),
1838     abjad.MarginMarkup(markup=abjad.Markup('bar.3'),),
1839     abjad.MarginMarkup(markup=abjad.Markup('bs.1'),),
1840     abjad.MarginMarkup(markup=abjad.Markup('bs.2'),),
1841     abjad.MarginMarkup(markup=abjad.Markup('cbs.'),),
1842 ])
1843
1844 names = cyc([
1845     abjad.StartMarkup(markup=abjad.Markup('Sopranino'),),
1846     abjad.StartMarkup(markup=abjad.Markup('Soprano 1'),),
1847     abjad.StartMarkup(markup=abjad.Markup('Soprano 2'),),
1848     abjad.StartMarkup(markup=abjad.Markup('Soprano 3'),),
1849     abjad.StartMarkup(markup=abjad.Markup('Alto 1'),),
1850     abjad.StartMarkup(markup=abjad.Markup('Alto 2'),),
1851     abjad.StartMarkup(markup=abjad.Markup('Alto 3'),),
1852     abjad.StartMarkup(markup=abjad.Markup('Alto 4'),),
1853     abjad.StartMarkup(markup=abjad.Markup('Alto 5'),),
1854     abjad.StartMarkup(markup=abjad.Markup('Alto 6'),),
1855     abjad.StartMarkup(markup=abjad.Markup('Tenor 1'),),
1856     abjad.StartMarkup(markup=abjad.Markup('Tenor 2'),),
1857     abjad.StartMarkup(markup=abjad.Markup('Tenor 3'),),
1858     abjad.StartMarkup(markup=abjad.Markup('Tenor 4'),),
1859     abjad.StartMarkup(markup=abjad.Markup('Tenor 5'),),
1860     abjad.StartMarkup(markup=abjad.Markup('Baritone 1'),),
1861     abjad.StartMarkup(markup=abjad.Markup('Baritone 2'),),
1862     abjad.StartMarkup(markup=abjad.Markup('Baritone 3'),),

```

```

1863 abjad.StartMarkup(markup=abjad.Markup('Bass 1'),),
1864 abjad.StartMarkup(markup=abjad.Markup('Bass 2'),),
1865 abjad.StartMarkup(markup=abjad.Markup('Contrabass'),),
1866 ])
1867
1868 for staff in abjad.iterate(score['Staff Group']).components(abjad.Staff):
1869     leaf1 = abjad.select(staff).leaves()[0]
1870     abjad.attach(next(instruments), leaf1)
1871     abjad.attach(next(abbreviations), leaf1)
1872     abjad.attach(next(names), leaf1)
1873
1874 for staff in abjad.select(score['Staff Group']).components(abjad.Staff):
1875     leaf1 = abjad.select(staff).leaves()[0]
1876     last_leaf = abjad.select(staff).leaves()[-1]
1877     abjad.attach(bar_line, last_leaf)
1878
1879 for staff in abjad.iterate(score['Global Context']).components(abjad.Staff):
1880     leaf1 = abjad.select(staff).leaves()[0]
1881     abjad.attach(mark, leaf1)
1882
1883 score_file = abjad.LilyPondFile.new(
1884     score,
1885     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1886     _stylesheets/abjad.ily'],
1887 )
1888
1889 abjad.SegmentMaker.comment_measure_numbers(score)
1890 ######
1891 directory = '/Users/evansdsg2/Scores/guerrero/Segments/Section_J'
1892 pdf_path = f'{directory}/Section_J.pdf'
1893 path = pathlib.Path('Section_J.pdf')
1894 if path.exists():
1895     print(f'Removing {pdf_path} ...')
1896     path.unlink()
1897 time_1 = time.time()
1898 print(f'Persisting {pdf_path} ...')
1899 result = abjad.persist(score_file).as_pdf(pdf_path)
1900 print(result[0])
1901 print(result[1])
1902 print(result[2])
1903 success = result[3]
1904 if success is False:
1905     print('LilyPond failed!')
1906 time_2 = time.time()
1907 total_time = time_2 - time_1
1908 print(f'Total time: {total_time} seconds')
1909 if path.exists():
1910     print(f'Opening {pdf_path} ...')
1911     os.system(f'open {pdf_path}')
1912 score_lines = open('/Users/evansdsg2/Scores/guerrero/Segments/Section_J/Section_J.
1913 ly').readlines()
1914 open('/Users/evansdsg2/Scores/guerrero/Build/Section_J.ly', 'w').writelines(
1915     score_lines[15:-1])
1916
1917 for staff in abjad.iterate(score['Staff 1']).components(abjad.Staff):
1918     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1919     signature_copy = abjad.mutate(signatures).copy()
1920     staff_copy = abjad.mutate(staff).copy()

```

```

1919 part = abjad.Score()
1920 part.insert(0, staff)
1921 part.insert(0, signature_copy)
1922 part_file = abjad.LilyPondFile.new(
1923     part,
1924     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1925 _stylesheets/abjad.ily'],
1926     )
1927 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/1.) soprano1'
1928 pdf_path = f'{directory}/Section_J.pdf'
1929 path = pathlib.Path('Section_J.pdf')
1930 if path.exists():
1931     print(f'Removing {pdf_path} ...')
1932     path.unlink()
1933 time_1 = time.time()
1934 print(f'Persisting {pdf_path} ...')
1935 result = abjad.persist(part_file).as_pdf(pdf_path)
1936 print(result[0])
1937 print(result[1])
1938 print(result[2])
1939 success = result[3]
1940 if success is False:
1941     print('LilyPond failed!')
1942 time_2 = time.time()
1943 total_time = time_2 - time_1
1944 print(f'Total time: {total_time} seconds')
1945 if path.exists():
1946     print(f'Opening {pdf_path} ...')
1947     os.system(f'open {pdf_path}')
1948 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.) soprano1/
1949 Section_J.ly').readlines()
1950 open('/Users/evansdsg2/Scores/guerrero/Build/parts/1.) soprano1/Section_J.ly',
1951 'w').writelines(part_lines[15:-1])
1952
1953 for staff in abjad.iterate(score['Staff 2']).components(abjad.Staff):
1954     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1955     signature_copy = abjad.mutate(signatures).copy()
1956     staff_copy = abjad.mutate(staff).copy()
1957     part = abjad.Score()
1958     part.insert(0, staff)
1959     part.insert(0, signature_copy)
1960     part_file = abjad.LilyPondFile.new(
1961         part,
1962         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1963 _stylesheets/abjad.ily'],
1964         )
1965 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/2.) soprano1'
1966 pdf_path = f'{directory}/Section_J.pdf'
1967 path = pathlib.Path('Section_J.pdf')
1968 if path.exists():
1969     print(f'Removing {pdf_path} ...')
1970     path.unlink()
1971 time_1 = time.time()
1972 print(f'Persisting {pdf_path} ...')
1973 result = abjad.persist(part_file).as_pdf(pdf_path)
1974 print(result[0])
1975 print(result[1])
1976 print(result[2])
1977 success = result[3]
```

```

1974     if success is False:
1975         print('LilyPond failed!')
1976     time_2 = time.time()
1977     total_time = time_2 - time_1
1978     print(f'Total time: {total_time} seconds')
1979     if path.exists():
1980         print(f'Opening {pdf_path} ...')
1981         os.system(f'open {pdf_path}')
1982     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/
1983     Section_J.ly').readlines()
1984     open('/Users/evansdsg2/Scores/guerrero/Build/parts/2.)soprano1/Section_J.ly',
1985     'w').writelines(part_lines[15:-1])
1986
1987     for staff in abjad.iterate(score['Staff 3']).components(abjad.Staff):
1988         signatures = abjad.select(score['Global Context']).components(abjad.Staff)
1989         signature_copy = abjad.mutate(signatures).copy()
1990         staff_copy = abjad.mutate(staff).copy()
1991         part = abjad.Score()
1992         part.insert(0, staff)
1993         part.insert(0, signature_copy)
1994         part_file = abjad.LilyPondFile.new(
1995             part,
1996             includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
1997             _stylesheets/abjad.ily'],
1998             )
1999         directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2'
2000         pdf_path = f'{directory}/Section_J.pdf'
2001         path = pathlib.Path('Section_J.pdf')
2002         if path.exists():
2003             print(f'Removing {pdf_path} ...')
2004             path.unlink()
2005         time_1 = time.time()
2006         print(f'Persisting {pdf_path} ...')
2007         result = abjad.persist(part_file).as_pdf(pdf_path)
2008         print(result[0])
2009         print(result[1])
2010         print(result[2])
2011         success = result[3]
2012         if success is False:
2013             print('LilyPond failed!')
2014         time_2 = time.time()
2015         total_time = time_2 - time_1
2016         print(f'Total time: {total_time} seconds')
2017         if path.exists():
2018             print(f'Opening {pdf_path} ...')
2019             os.system(f'open {pdf_path}')
2020             part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/
2021             Section_J.ly').readlines()
2022             open('/Users/evansdsg2/Scores/guerrero/Build/parts/3.)soprano2/Section_J.ly',
2023             'w').writelines(part_lines[15:-1])
2024
2025             for staff in abjad.iterate(score['Staff 4']).components(abjad.Staff):
2026                 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2027                 signature_copy = abjad.mutate(signatures).copy()

```

```

2028     part,
2029     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2030 _stylesheets/abjad.ily'],
2031     )
2032 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3'
2033 pdf_path = f'{directory}/Section_J.pdf'
2034 path = pathlib.Path('Section_J.pdf')
2035 if path.exists():
2036     print(f'Removing {pdf_path} ...')
2037     path.unlink()
2038 time_1 = time.time()
2039 print(f'Persisting {pdf_path} ...')
2040 result = abjad.persist(part_file).as_pdf(pdf_path)
2041 print(result[0])
2042 print(result[1])
2043 print(result[2])
2044 success = result[3]
2045 if success is False:
2046     print('LilyPond failed!')
2047 time_2 = time.time()
2048 total_time = time_2 - time_1
2049 print(f'Total time: {total_time} seconds')
2050 if path.exists():
2051     print(f'Opening {pdf_path} ...')
2052     os.system(f'open {pdf_path}')
2053 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/
2054 Section_J.ly').readlines()
2055 open('/Users/evansdsg2/Scores/guerrero/Build/parts/4.)soprano3/Section_J.ly',
2056      'w').writelines(part_lines[15:-1])
2057
2058 for staff in abjad.iterate(score['Staff 5']).components(abjad.Staff):
2059     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2060     signature_copy = abjad.mutate(signatures).copy()
2061     staff_copy = abjad.mutate(staff).copy()
2062     part = abjad.Score()
2063     part.insert(0, staff)
2064     part.insert(0, signature_copy)
2065     part_file = abjad.LilyPondFile.new(
2066         part,
2067         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2068 _stylesheets/abjad.ily'],
2069         )
2070 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/5.)alto1'
2071 pdf_path = f'{directory}/Section_J.pdf'
2072 path = pathlib.Path('Section_J.pdf')
2073 if path.exists():
2074     print(f'Removing {pdf_path} ...')
2075     path.unlink()
2076 time_1 = time.time()
2077 print(f'Persisting {pdf_path} ...')
2078 result = abjad.persist(part_file).as_pdf(pdf_path)
2079 print(result[0])
2080 print(result[1])
2081 print(result[2])
2082 success = result[3]
2083 if success is False:
2084     print('LilyPond failed!')
2085 time_2 = time.time()
2086 total_time = time_2 - time_1

```

```

2083     print(f'Total time: {total_time} seconds')
2084     if path.exists():
2085         print(f'Opening {pdf_path} ...')
2086         os.system(f'open {pdf_path}')
2087     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.')alto1/
2088     Section_J.ly').readlines()
2089     open('/Users/evansdsg2/Scores/guerrero/Build/parts/5.')alto1/Section_J.ly', 'w'
2090     ).writelines(part_lines[15:-1])
2091
2092 for staff in abjad.iterate(score['Staff 6']).components(abjad.Staff):
2093     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2094     signature_copy = abjad.mutate(signatures).copy()
2095     staff_copy = abjad.mutate(staff).copy()
2096     part = abjad.Score()
2097     part.insert(0, staff)
2098     part.insert(0, signature_copy)
2099     part_file = abjad.LilyPondFile.new(
2100         part,
2101         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2102         _stylesheets/abjad.ily'],
2103         )
2104     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/6.')alto2'
2105     pdf_path = f'{directory}/Section_J.pdf'
2106     path = pathlib.Path('Section_J.pdf')
2107     if path.exists():
2108         print(f'Removing {pdf_path} ...')
2109         path.unlink()
2110     time_1 = time.time()
2111     print(f'Persisting {pdf_path} ...')
2112     result = abjad.persist(part_file).as_pdf(pdf_path)
2113     print(result[0])
2114     print(result[1])
2115     print(result[2])
2116     success = result[3]
2117     if success is False:
2118         print('LilyPond failed!')
2119     time_2 = time.time()
2120     total_time = time_2 - time_1
2121     print(f'Total time: {total_time} seconds')
2122     if path.exists():
2123         print(f'Opening {pdf_path} ...')
2124         os.system(f'open {pdf_path}')
2125     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.')alto2/
2126     Section_J.ly').readlines()
2127     open('/Users/evansdsg2/Scores/guerrero/Build/parts/6.')alto2/Section_J.ly', 'w'
2128     ).writelines(part_lines[15:-1])
2129
2130 for staff in abjad.iterate(score['Staff 7']).components(abjad.Staff):
2131     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2132     signature_copy = abjad.mutate(signatures).copy()
2133     staff_copy = abjad.mutate(staff).copy()
2134     part = abjad.Score()
2135     part.insert(0, staff)
2136     part.insert(0, signature_copy)
2137     part_file = abjad.LilyPondFile.new(
2138         part,
2139         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2140         _stylesheets/abjad.ily'],
2141         )

```

```

2136 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3'
2137 pdf_path = f'{directory}/Section_J.pdf'
2138 path = pathlib.Path('Section_J.pdf')
2139 if path.exists():
2140     print(f'Removing {pdf_path} ...')
2141     path.unlink()
2142 time_1 = time.time()
2143 print(f'Persisting {pdf_path} ...')
2144 result = abjad.persist(part_file).as_pdf(pdf_path)
2145 print(result[0])
2146 print(result[1])
2147 print(result[2])
2148 success = result[3]
2149 if success is False:
2150     print('LilyPond failed!')
2151 time_2 = time.time()
2152 total_time = time_2 - time_1
2153 print(f'Total time: {total_time} seconds')
2154 if path.exists():
2155     print(f'Opening {pdf_path} ...')
2156     os.system(f'open {pdf_path}')
2157 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/
Section_J.ly').readlines()
2158 open('/Users/evansdsg2/Scores/guerrero/Build/parts/7.)alto3/Section_J.ly', 'w'
).writelines(part_lines[15:-1])
2159
2160 for staff in abjad.iterate(score['Staff 8']).components(abjad.Staff):
2161     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2162     signature_copy = abjad.mutate(signatures).copy()
2163     staff_copy = abjad.mutate(staff).copy()
2164     part = abjad.Score()
2165     part.insert(0, staff)
2166     part.insert(0, signature_copy)
2167     part_file = abjad.LilyPondFile.new(
2168         part,
2169         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2170         )
2171 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4'
2172 pdf_path = f'{directory}/Section_J.pdf'
2173 path = pathlib.Path('Section_J.pdf')
2174 if path.exists():
2175     print(f'Removing {pdf_path} ...')
2176     path.unlink()
2177 time_1 = time.time()
2178 print(f'Persisting {pdf_path} ...')
2179 result = abjad.persist(part_file).as_pdf(pdf_path)
2180 print(result[0])
2181 print(result[1])
2182 print(result[2])
2183 success = result[3]
2184 if success is False:
2185     print('LilyPond failed!')
2186 time_2 = time.time()
2187 total_time = time_2 - time_1
2188 print(f'Total time: {total_time} seconds')
2189 if path.exists():
2190     print(f'Opening {pdf_path} ...')
2191     os.system(f'open {pdf_path}')

```

```

2192 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/
2193 Section_J.ly').readlines()
2194 open('/Users/evansdsg2/Scores/guerrero/Build/parts/8.)alto4/Section_J.ly', 'w'
2195 ).writelines(part_lines[15:-1])
2196
2197 for staff in abjad.iterate(score['Staff 9']).components(abjad.Staff):
2198 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2199 signature_copy = abjad.mutate(signatures).copy()
2200 staff_copy = abjad.mutate(staff).copy()
2201 part = abjad.Score()
2202 part.insert(0, staff)
2203 part.insert(0, signature_copy)
2204 part_file = abjad.LilyPondFile.new(
2205     part,
2206     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2207 _stylesheets/abjad.ily'],
2208 )
2209 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5'
2210 pdf_path = f'{directory}/Section_J.pdf'
2211 path = pathlib.Path('Section_J.pdf')
2212 if path.exists():
2213     print(f'Removing {pdf_path} ...')
2214     path.unlink()
2215 time_1 = time.time()
2216 print(f'Persisting {pdf_path} ...')
2217 result = abjad.persist(part_file).as_pdf(pdf_path)
2218 print(result[0])
2219 print(result[1])
2220 print(result[2])
2221 success = result[3]
2222 if success is False:
2223     print('LilyPond failed!')
2224 time_2 = time.time()
2225 total_time = time_2 - time_1
2226 print(f'Total time: {total_time} seconds')
2227 if path.exists():
2228     print(f'Opening {pdf_path} ...')
2229     os.system(f'open {pdf_path}')
2230 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/
2231 Section_J.ly').readlines()
2232 open('/Users/evansdsg2/Scores/guerrero/Build/parts/9.)alto5/Section_J.ly', 'w'
2233 ).writelines(part_lines[15:-1])
2234
2235 for staff in abjad.iterate(score['Staff 10']).components(abjad.Staff):
2236 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2237 signature_copy = abjad.mutate(signatures).copy()
2238 staff_copy = abjad.mutate(staff).copy()
2239 part = abjad.Score()
2240 part.insert(0, staff)
2241 part.insert(0, signature_copy)
2242 part_file = abjad.LilyPondFile.new(
2243     part,
2244     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2245 _stylesheets/abjad.ily'],
2246 )
2247 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6'
2248 pdf_path = f'{directory}/Section_J.pdf'
2249 path = pathlib.Path('Section_J.pdf')
2250 if path.exists():

```

```

2245     print(f'Removing {pdf_path} ...')
2246     path.unlink()
2247 time_1 = time.time()
2248 print(f'Persisting {pdf_path} ...')
2249 result = abjad.persist(part_file).as_pdf(pdf_path)
2250 print(result[0])
2251 print(result[1])
2252 print(result[2])
2253 success = result[3]
2254 if success is False:
2255     print('LilyPond failed!')
2256 time_2 = time.time()
2257 total_time = time_2 - time_1
2258 print(f'Total time: {total_time} seconds')
2259 if path.exists():
2260     print(f'Opening {pdf_path} ...')
2261     os.system(f'open {pdf_path}')
2262 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/
Section_J.ly').readlines()
2263 open('/Users/evansdsg2/Scores/guerrero/Build/parts/10.)alto6/Section_J.ly', 'w'
').writelines(part_lines[15:-1])
2264
2265 for staff in abjad.iterate(score['Staff 11']).components(abjad.Staff):
2266 signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2267 signature_copy = abjad.mutate(signatures).copy()
2268 staff_copy = abjad.mutate(staff).copy()
2269 part = abjad.Score()
2270 part.insert(0, staff)
2271 part.insert(0, signature_copy)
2272 part_file = abjad.LilyPondFile.new(
2273     part,
2274     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
_stylesheets/abjad.ily'],
2275     )
2276 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1'
2277 pdf_path = f'{directory}/Section_J.pdf'
2278 path = pathlib.Path('Section_J.pdf')
2279 if path.exists():
2280     print(f'Removing {pdf_path} ...')
2281     path.unlink()
2282 time_1 = time.time()
2283 print(f'Persisting {pdf_path} ...')
2284 result = abjad.persist(part_file).as_pdf(pdf_path)
2285 print(result[0])
2286 print(result[1])
2287 print(result[2])
2288 success = result[3]
2289 if success is False:
2290     print('LilyPond failed!')
2291 time_2 = time.time()
2292 total_time = time_2 - time_1
2293 print(f'Total time: {total_time} seconds')
2294 if path.exists():
2295     print(f'Opening {pdf_path} ...')
2296     os.system(f'open {pdf_path}')
2297 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/
Section_J.ly').readlines()
2298 open('/Users/evansdsg2/Scores/guerrero/Build/parts/11.)tenor1/Section_J.ly', 'w'
').writelines(part_lines[15:-1])

```

```

2299
2300 for staff in abjad.iterate(score['Staff 12']).components(abjad.Staff):
2301     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2302     signature_copy = abjad.mutate(signatures).copy()
2303     staff_copy = abjad.mutate(staff).copy()
2304     part = abjad.Score()
2305     part.insert(0, staff)
2306     part.insert(0, signature_copy)
2307     part_file = abjad.LilyPondFile.new(
2308         part,
2309         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2310         _stylesheets/abjad.ily'],
2311         )
2312     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2'
2313     pdf_path = f'{directory}/Section_J.pdf'
2314     path = pathlib.Path('Section_J.pdf')
2315     if path.exists():
2316         print(f'Removing {pdf_path} ...')
2317         path.unlink()
2318     time_1 = time.time()
2319     print(f'Persisting {pdf_path} ...')
2320     result = abjad.persist(part_file).as_pdf(pdf_path)
2321     print(result[0])
2322     print(result[1])
2323     print(result[2])
2324     success = result[3]
2325     if success is False:
2326         print('LilyPond failed!')
2327     time_2 = time.time()
2328     total_time = time_2 - time_1
2329     print(f'Total time: {total_time} seconds')
2330     if path.exists():
2331         print(f'Opening {pdf_path} ...')
2332         os.system(f'open {pdf_path}')
2333     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/
2334     Section_J.ly').readlines()
2335     open('/Users/evansdsg2/Scores/guerrero/Build/parts/12.)tenor2/Section_J.ly', ,
2336     'w').writelines(part_lines[15:-1])
2337
2338 for staff in abjad.iterate(score['Staff 13']).components(abjad.Staff):
2339     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2340     signature_copy = abjad.mutate(signatures).copy()
2341     staff_copy = abjad.mutate(staff).copy()
2342     part = abjad.Score()
2343     part.insert(0, staff)
2344     part.insert(0, signature_copy)
2345     part_file = abjad.LilyPondFile.new(
2346         part,
2347         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2348         _stylesheets/abjad.ily'],
2349         )
2350     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/13.)tenor3'
2351     pdf_path = f'{directory}/Section_J.pdf'
2352     path = pathlib.Path('Section_J.pdf')
2353     if path.exists():
2354         print(f'Removing {pdf_path} ...')
2355         path.unlink()
2356     time_1 = time.time()
2357     print(f'Persisting {pdf_path} ...')

```

```

2354     result = abjad.persist(part_file).as_pdf(pdf_path)
2355     print(result[0])
2356     print(result[1])
2357     print(result[2])
2358     success = result[3]
2359     if success is False:
2360         print('LilyPond failed!')
2361     time_2 = time.time()
2362     total_time = time_2 - time_1
2363     print(f'Total time: {total_time} seconds')
2364     if path.exists():
2365         print(f'Opening {pdf_path} ...')
2366         os.system(f'open {pdf_path}')
2367     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/
2368     Section_J.ly').readlines()
2369     open('/Users/evansdsg2/Scores/guerrero/Build/parts/13.')tenor3/Section_J.ly', ,
2370     w').writelines(part_lines[15:-1])
2371
2372 for staff in abjad.iterate(score['Staff 14']).components(abjad.Staff):
2373     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2374     signature_copy = abjad.mutate(signatures).copy()
2375     staff_copy = abjad.mutate(staff).copy()
2376     part = abjad.Score()
2377     part.insert(0, staff)
2378     part.insert(0, signature_copy)
2379     part_file = abjad.LilyPondFile.new(
2380         part,
2381         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2382         _stylesheets/abjad.ily'],
2383         )
2384     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4'
2385     pdf_path = f'{directory}/Section_J.pdf'
2386     path = pathlib.Path('Section_J.pdf')
2387     if path.exists():
2388         print(f'Removing {pdf_path} ...')
2389         path.unlink()
2390     time_1 = time.time()
2391     print(f'Persisting {pdf_path} ...')
2392     result = abjad.persist(part_file).as_pdf(pdf_path)
2393     print(result[0])
2394     print(result[1])
2395     print(result[2])
2396     success = result[3]
2397     if success is False:
2398         print('LilyPond failed!')
2399     time_2 = time.time()
2400     total_time = time_2 - time_1
2401     print(f'Total time: {total_time} seconds')
2402     if path.exists():
2403         print(f'Opening {pdf_path} ...')
2404         os.system(f'open {pdf_path}')
2405     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/
2406     Section_J.ly').readlines()
2407     open('/Users/evansdsg2/Scores/guerrero/Build/parts/14.')tenor4/Section_J.ly', ,
2408     w').writelines(part_lines[15:-1])
2409
2410 for staff in abjad.iterate(score['Staff 15']).components(abjad.Staff):
2411     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2412     signature_copy = abjad.mutate(signatures).copy()

```

```

2408 staff_copy = abjad.mutate(staff).copy()
2409 part = abjad.Score()
2410 part.insert(0, staff)
2411 part.insert(0, signature_copy)
2412 part_file = abjad.LilyPondFile.new(
2413     part,
2414     includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2415 _stylesheets/abjad.ily'],
2416 )
2417 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5'
2418 pdf_path = f'{directory}/Section_J.pdf'
2419 path = pathlib.Path('Section_J.pdf')
2420 if path.exists():
2421     print(f'Removing {pdf_path} ...')
2422     path.unlink()
2423 time_1 = time.time()
2424 print(f'Persisting {pdf_path} ...')
2425 result = abjad.persist(part_file).as_pdf(pdf_path)
2426 print(result[0])
2427 print(result[1])
2428 print(result[2])
2429 success = result[3]
2430 if success is False:
2431     print('LilyPond failed!')
2432 time_2 = time.time()
2433 total_time = time_2 - time_1
2434 print(f'Total time: {total_time} seconds')
2435 if path.exists():
2436     print(f'Opening {pdf_path} ...')
2437     os.system(f'open {pdf_path}')
2438 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/15.')tenor5/
2439 Section_J.ly').readlines()
2440 for staff in abjad.iterate(score['Staff 16']).components(abjad.Staff):
2441     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2442     signature_copy = abjad.mutate(signatures).copy()
2443     staff_copy = abjad.mutate(staff).copy()
2444     part = abjad.Score()
2445     part.insert(0, staff)
2446     part.insert(0, signature_copy)
2447     part_file = abjad.LilyPondFile.new(
2448         part,
2449         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2450 _stylesheets/abjad.ily'],
2451     )
2452     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1'
2453     pdf_path = f'{directory}/Section_J.pdf'
2454     path = pathlib.Path('Section_J.pdf')
2455     if path.exists():
2456         print(f'Removing {pdf_path} ...')
2457         path.unlink()
2458     time_1 = time.time()
2459     print(f'Persisting {pdf_path} ...')
2460     result = abjad.persist(part_file).as_pdf(pdf_path)
2461     print(result[0])
2462     print(result[1])
2463     print(result[2])

```

```

2463     success = result[3]
2464     if success is False:
2465         print('LilyPond failed!')
2466     time_2 = time.time()
2467     total_time = time_2 - time_1
2468     print(f'Total time: {total_time} seconds')
2469     if path.exists():
2470         print(f'Opening {pdf_path} ...')
2471         os.system(f'open {pdf_path}')
2472     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/
2473     Section_J.ly').readlines()
2474     open('/Users/evansdsg2/Scores/guerrero/Build/parts/16.')baritone1/Section_J.ly',
2475     'w').writelines(part_lines[15:-1])
2476
2477 for staff in abjad.iterate(score['Staff 17']).components(abjad.Staff):
2478     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2479     signature_copy = abjad.mutate(signatures).copy()
2480     staff_copy = abjad.mutate(staff).copy()
2481     part = abjad.Score()
2482     part.insert(0, staff)
2483     part.insert(0, signature_copy)
2484     part_file = abjad.LilyPondFile.new(
2485         part,
2486         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2487         _stylesheets/abjad.ily'],
2488         )
2489     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2'
2490     pdf_path = f'{directory}/Section_J.pdf'
2491     path = pathlib.Path('Section_J.pdf')
2492     if path.exists():
2493         print(f'Removing {pdf_path} ...')
2494         path.unlink()
2495     time_1 = time.time()
2496     print(f'Persisting {pdf_path} ...')
2497     result = abjad.persist(part_file).as_pdf(pdf_path)
2498     print(result[0])
2499     print(result[1])
2500     print(result[2])
2501     success = result[3]
2502     if success is False:
2503         print('LilyPond failed!')
2504     time_2 = time.time()
2505     total_time = time_2 - time_1
2506     print(f'Total time: {total_time} seconds')
2507     if path.exists():
2508         print(f'Opening {pdf_path} ...')
2509         os.system(f'open {pdf_path}')
2510     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/
2511     Section_J.ly').readlines()
2512     open('/Users/evansdsg2/Scores/guerrero/Build/parts/17.')baritone2/Section_J.ly',
2513     'w').writelines(part_lines[15:-1])
2514
2515 for staff in abjad.iterate(score['Staff 18']).components(abjad.Staff):
2516     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2517     signature_copy = abjad.mutate(signatures).copy()
2518     staff_copy = abjad.mutate(staff).copy()
2519     part = abjad.Score()
2520     part.insert(0, staff)
2521     part.insert(0, signature_copy)

```

```

2517     part_file = abjad.LilyPondFile.new(
2518         part,
2519         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2520         _stylesheets/abjad.ily'],
2521         )
2521 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3'
2522 pdf_path = f'{directory}/Section_J.pdf'
2523 path = pathlib.Path('Section_J.pdf')
2524 if path.exists():
2525     print(f'Removing {pdf_path} ...')
2526     path.unlink()
2527 time_1 = time.time()
2528 print(f'Persisting {pdf_path} ...')
2529 result = abjad.persist(part_file).as_pdf(pdf_path)
2530 print(result[0])
2531 print(result[1])
2532 print(result[2])
2533 success = result[3]
2534 if success is False:
2535     print('LilyPond failed!')
2536 time_2 = time.time()
2537 total_time = time_2 - time_1
2538 print(f'Total time: {total_time} seconds')
2539 if path.exists():
2540     print(f'Opening {pdf_path} ...')
2541     os.system(f'open {pdf_path}')
2542 part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/
2542 Section_J.ly').readlines()
2543 open('/Users/evansdsg2/Scores/guerrero/Build/parts/18.)baritone3/Section_J.ly',
2543 , 'w').writelines(part_lines[15:-1])
2544
2545 for staff in abjad.iterate(score['Staff 19']).components(abjad.Staff):
2546     signatures = abjad.select(score['Global Context']).components(abjad.Staff)
2547     signature_copy = abjad.mutate(signatures).copy()
2548     staff_copy = abjad.mutate(staff).copy()
2549     part = abjad.Score()
2550     part.insert(0, staff)
2551     part.insert(0, signature_copy)
2552     part_file = abjad.LilyPondFile.new(
2553         part,
2554         includes=['first_stylesheet.ily', '/Users/evansdsg2/abjad/docs/source/
2554         _stylesheets/abjad.ily'],
2555         )
2556 directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1'
2557 pdf_path = f'{directory}/Section_J.pdf'
2558 path = pathlib.Path('Section_J.pdf')
2559 if path.exists():
2560     print(f'Removing {pdf_path} ...')
2561     path.unlink()
2562 time_1 = time.time()
2563 print(f'Persisting {pdf_path} ...')
2564 result = abjad.persist(part_file).as_pdf(pdf_path)
2565 print(result[0])
2566 print(result[1])
2567 print(result[2])
2568 success = result[3]
2569 if success is False:
2570     print('LilyPond failed!')
2571 time_2 = time.time()

```

```

2572     total_time = time_2 - time_1
2573     print(f'Total time: {total_time} seconds')
2574     if path.exists():
2575         print(f'Opening {pdf_path} ...')
2576         os.system(f'open {pdf_path}')
2577     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/
2578     Section_J.ly').readlines()
2579     open('/Users/evansdsg2/Scores/guerrero/Build/parts/19.)bass1/Section_J.ly', 'w'
2580     ).writelines(part_lines[15:-1])
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624

```

```
2625     )
2626     directory = '/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
2627     pdf_path = f'{directory}/Section_J.pdf'
2628     path = pathlib.Path('Section_J.pdf')
2629     if path.exists():
2630         print(f'Removing {pdf_path} ...')
2631         path.unlink()
2632     time_1 = time.time()
2633     print(f'Persisting {pdf_path} ...')
2634     result = abjad.persist(part_file).as_pdf(pdf_path)
2635     print(result[0])
2636     print(result[1])
2637     print(result[2])
2638     success = result[3]
2639     if success is False:
2640         print('LilyPond failed!')
2641     time_2 = time.time()
2642     total_time = time_2 - time_1
2643     print(f'Total time: {total_time} seconds')
2644     if path.exists():
2645         print(f'Opening {pdf_path} ...')
2646         os.system(f'open {pdf_path}')
2647     part_lines = open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass'
2648     /Section_J.ly').readlines()
2649     open('/Users/evansdsg2/Scores/guerrero/Build/parts/21.)contrabass/Section_J.ly',
2650     'w').writelines(part_lines[15:-1])
```

Listing 3.11: Invocation Source Code



## **Chapter 4**

### **Score**

# GUERRERO

for 21 saxophones

2018

Gregory Rowland Evans

## FOREWORD

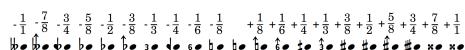
*GUERRERO* is a Spanish word that means "Warrior." This piece is intended, in part, to be an homage to the surname of composer Francisco Guerrero Marin and to his work *Rhea* for twelve saxophones, but the piece is primarily inspired by the figurative notions of reflection, refraction, and illumination. *GUERRERO* begins with an *Invocation* to its musical forerunner and to the muses of the epic mythology that led to the names of many stars, planets, and moons, before setting off on a journey of electric metamorphosis. (G.R.E.)

Sailing on an ocean of time and memory, the Warrior matches measured combinations with breath and aether, neither hot nor cold, neither wet nor dry. Where once there was nothing is now the electromagnetic scintillation of particulate light, pulsing through the circulation where sight is now known. Whirlpools of stellar, parallactic aberration help mark the distance as the Warrior drifts away. The iron sea is a preface to a violent birth. (G.R.E.)

## PERFORMANCE NOTES

Score is transposed.

### Microtones:



Accidentals apply only to the pitch which they immediately precede, but persist through ties. Microtones may be achieved either through the embouchure or fingerings.

The symbol “◦” over a note represents a mostly airy tone-color that still retains some pitch.

The symbol “ø” represents a tone-color that is halfway between a normal playing technique and the ◦ technique.

A “+” over a note indicates a “tongue slap” or “tongue pizz.” technique.

## INSTRUMENTATION

1 Soprano Saxophone

3 Soprano Saxophones

6 Alto Saxophones

5 Tenor Saxophones

3 Baritone Saxophones

2 Bass Saxophones

1 Contrabass Saxophone

c.5'30"

to the Frost Saxophone Ensemble  
**GUERRERO**  
 for twenty-one saxophones

Gregory Rowland Evans

**5** *J = 60* **Invocation**

**4**

**3**

Soprano

Soprano 1

Soprano 2

Soprano 3

Alto 1

Alto 2

Alto 3

Alto 4

Alto 5

Alto 6

Tenor 1

Tenor 2

Tenor 3

Tenor 4

Tenor 5

Bariitone 1

Bariitone 2

Bariitone 3

Base 1

Base 2

Contrabass

GUERRERO - I - Evans

④ **2**

spn.  
spn.  
spn.  
spn.  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
alt.6  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bar.1  
bar.2  
bar.3  
ba.1  
ba.2  
ba.

**4**

spn.  
spn.  
spn.  
spn.  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
alt.6  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bar.1  
bar.2  
bar.3  
ba.1  
ba.2  
ba.

**3**

spn.  
spn.  
spn.  
spn.  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
alt.6  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bar.1  
bar.2  
bar.3  
ba.1  
ba.2  
ba.

⑦ **3**

**4**

**5**

GUERRERO -3- Evans

(10) **3** **4** **9**

spn.1  
spn.2  
spn.3  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
alt.6  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bass.1  
bass.2  
bass.3  
bass.4  
cha.

(13)  5 [A]

44



spn.  
asp.1  
asp.2  
asp.3  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bas.1  
bas.2  
bas.3  
bas.4  
pno.

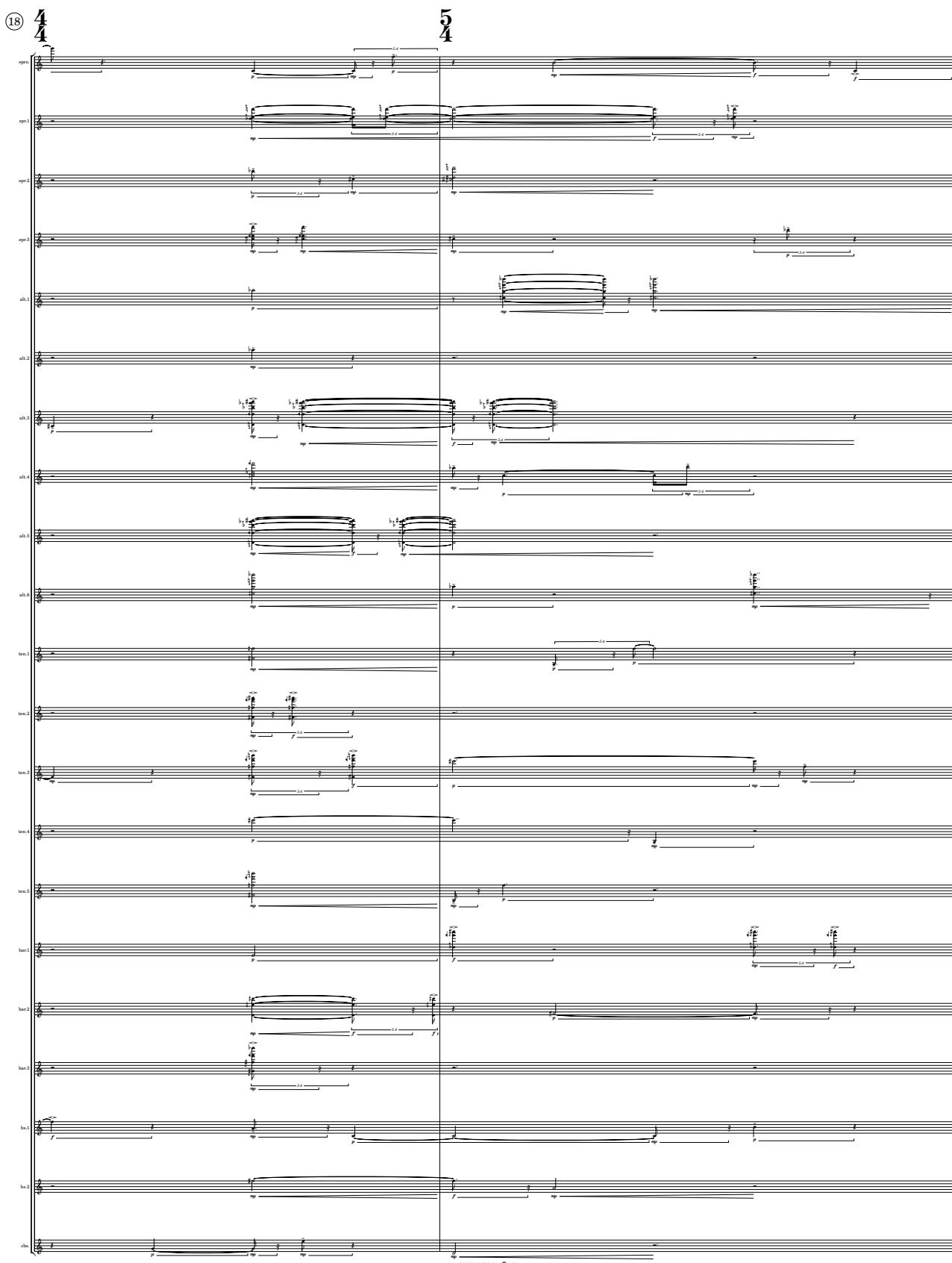
GUERRERO -5- Evans

(15) **3**

**4**

**3**

GUERRERO -6- Evans

(18) 

(20)

(22) **3**

**4**

**3**

GUERRERO -9- Evans

(25) **4**

**5**

GUERRERO -10- Evans

(27) **5**

**4**

GUERRERO -11- Evans

(29)

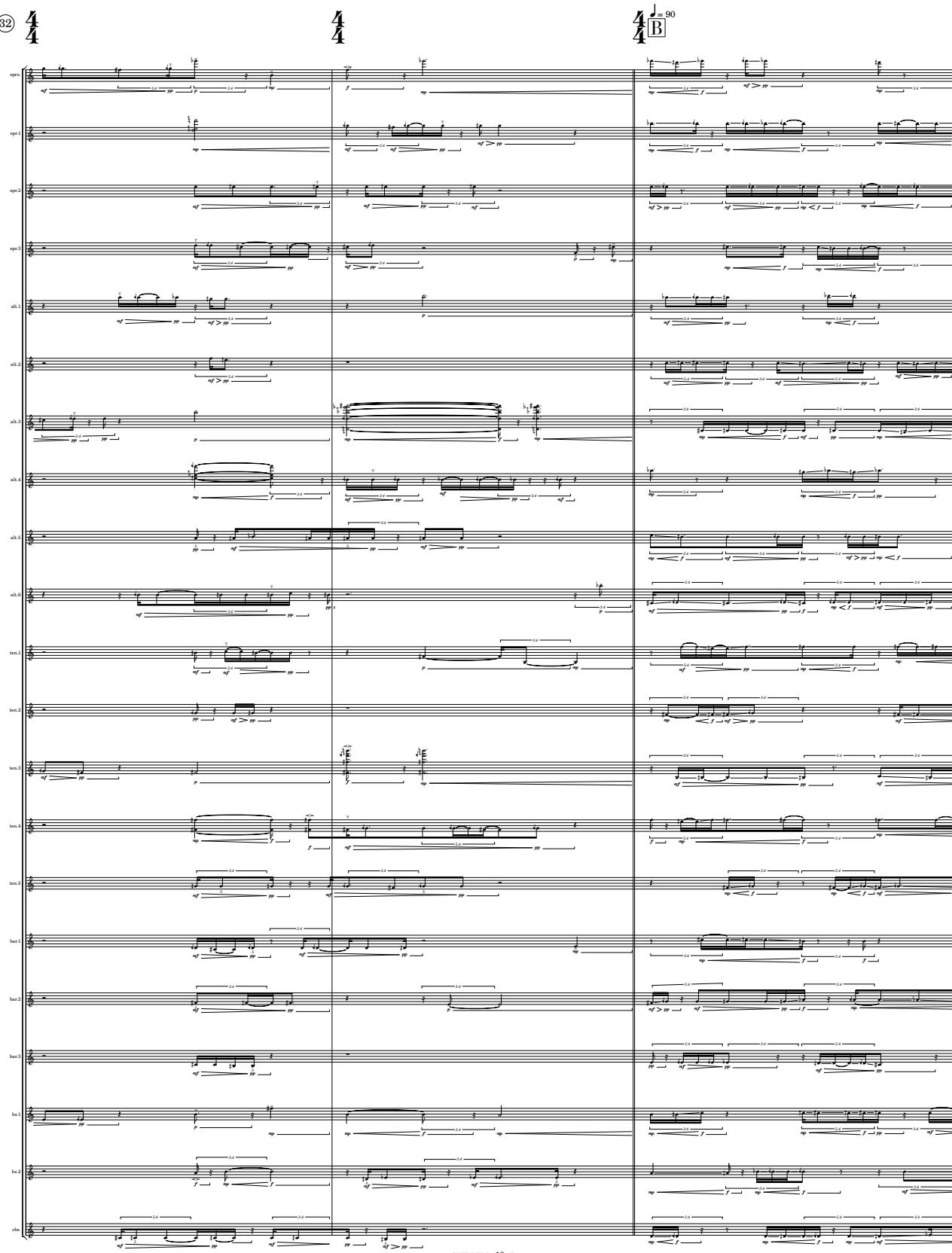
**3**

**4**

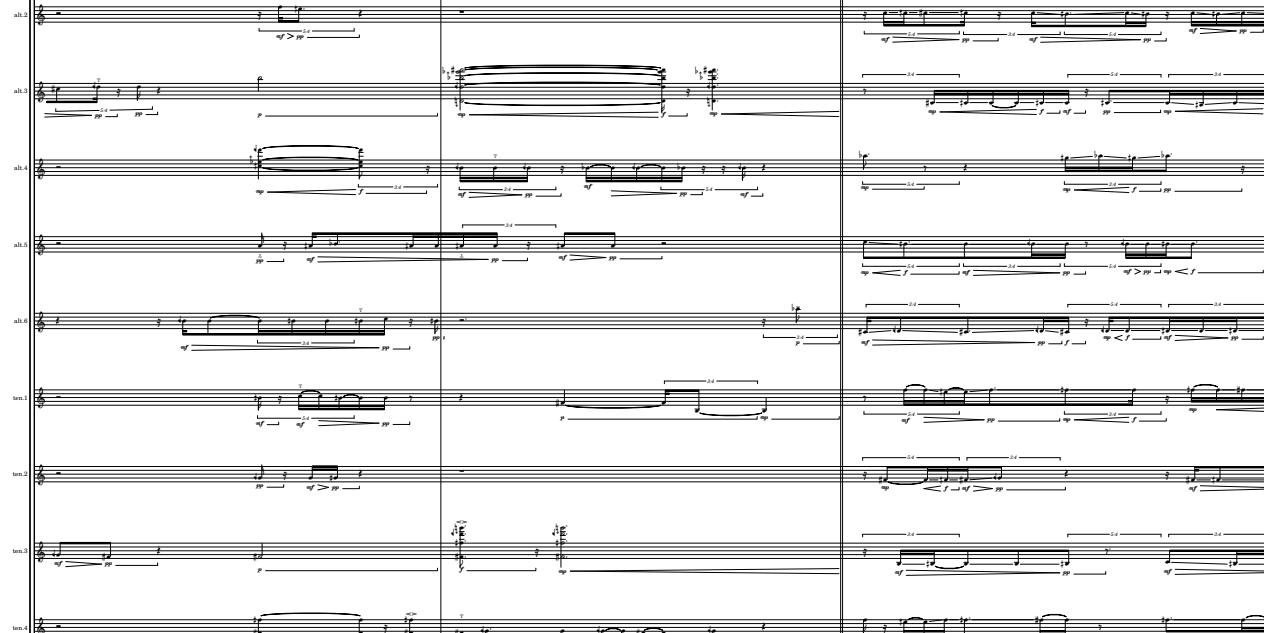
**3**

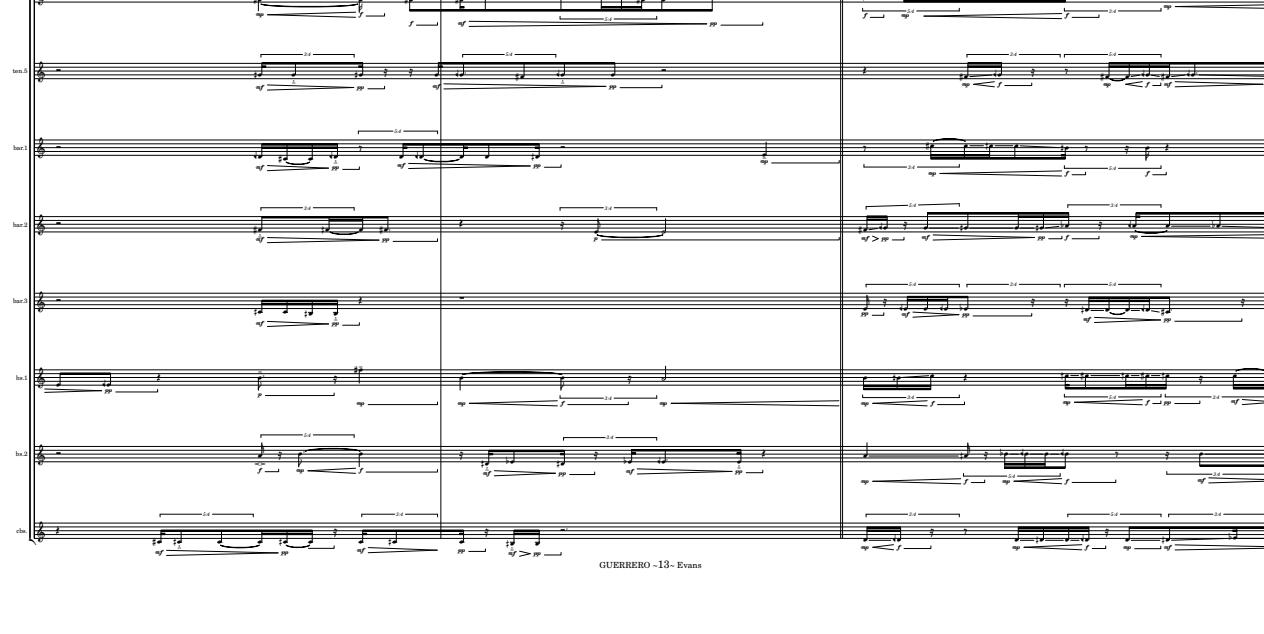
spn.  
sop.1  
sop.2  
sop.3  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bar.1  
bar.2  
bar.3  
bar.4  
tuba

GUERRERO -12- Evans

(32) 

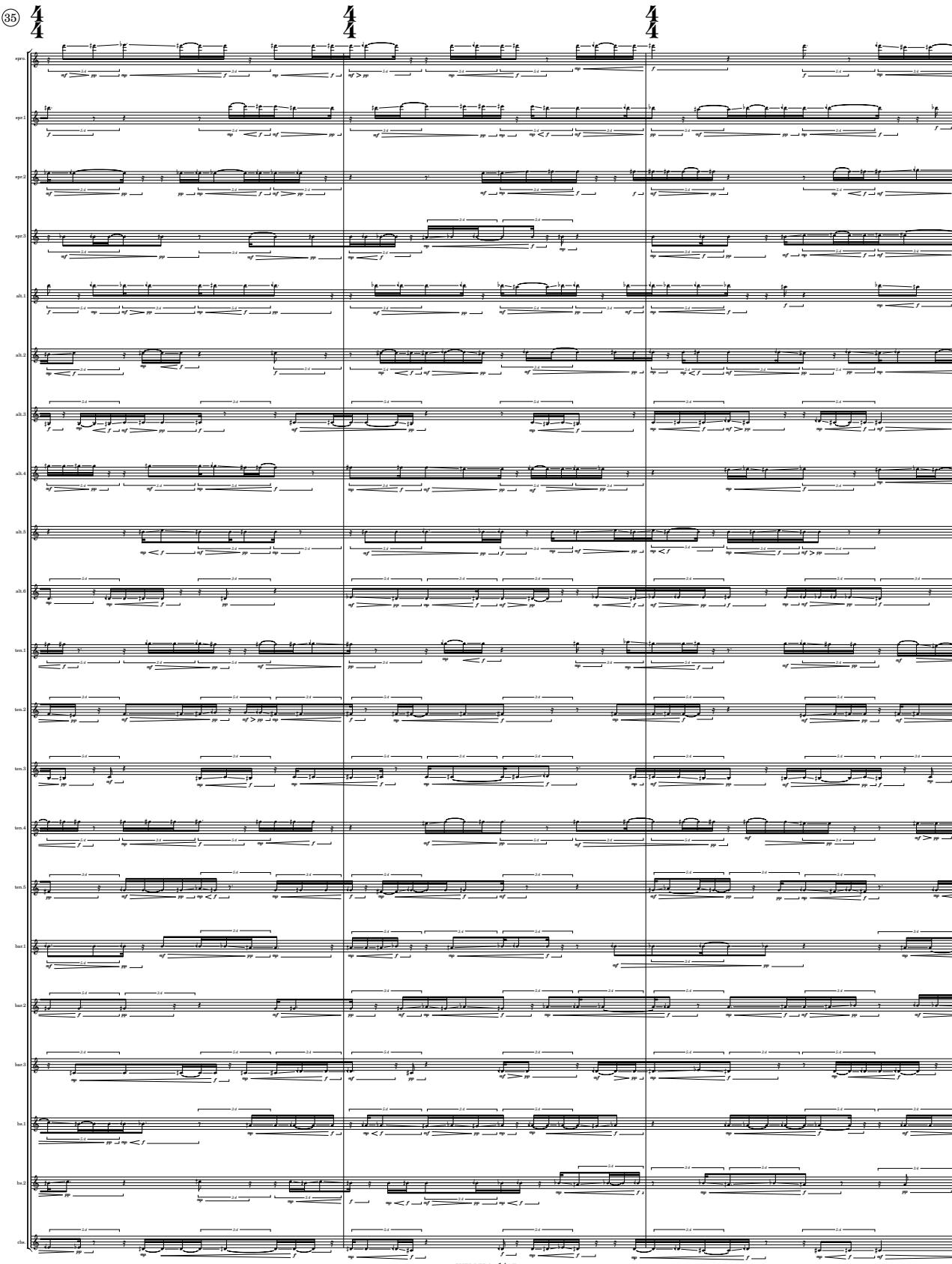






CHAPTER 4. SCORE

GUERRERO -13- Evans

(35) 

The musical score page contains three staves of music for a large ensemble. The instruments listed on the left are: spt.1, spt.2, spt.3, alt.1, alt.2, alt.3, alt.4, alt.5, alt.6, ten.1, ten.2, ten.3, ten.4, ten.5, bar.1, bar.2, bas.1, bas.2, bas.3, bcl.1, bcl.2, and cl. The score is divided into three measures. Measure 1 consists of three staves. Measure 2 consists of two staves. Measure 3 consists of three staves. The notation includes various dynamics such as  $\text{mf}$ ,  $\text{pp}$ ,  $\text{f}$ ,  $\text{ff}$ , and  $\text{ff}$ . Measure 1 starts with  $\text{mf}$  dynamics for spt.1, spt.2, spt.3, alt.1, alt.2, alt.3, alt.4, alt.5, alt.6, ten.1, ten.2, ten.3, ten.4, ten.5, bar.1, bar.2, and bas.1. It ends with  $\text{f}$  dynamics for all instruments. Measure 2 starts with  $\text{f}$  dynamics for spt.1, spt.2, spt.3, alt.1, alt.2, alt.3, alt.4, alt.5, alt.6, ten.1, ten.2, ten.3, ten.4, ten.5, bar.1, bar.2, and bas.1. It ends with  $\text{ff}$  dynamics for all instruments. Measure 3 starts with  $\text{ff}$  dynamics for spt.1, spt.2, spt.3, alt.1, alt.2, alt.3, alt.4, alt.5, alt.6, ten.1, ten.2, ten.3, ten.4, ten.5, bar.1, bar.2, and bas.1. It ends with  $\text{f}$  dynamics for all instruments.

GUERRERO -14- Evans

(38) 

The musical score consists of three staves of music for orchestra and piano. The first staff includes parts for soprano, alto, tenor, bass, and piano. The second staff includes parts for flute, oboe, clarinet, bassoon, and piano. The third staff includes parts for trumpet, horn, and piano. The score is in common time (indicated by a '4'). Measure 1 starts with a dynamic of  $\text{mf}$ . Measures 2-3 show various dynamics including  $\text{f}$ ,  $\text{pp}$ , and  $\text{mf} > \text{pp}$ . The piano part features sustained notes and chords throughout the section.



(44) 

The musical score consists of three staves of music for orchestra and piano. The first staff includes the piano (pno), soprano (sopr), alto 1 (alt.1), alto 2 (alt.2), alto 3 (alt.3), alto 4 (alt.4), alto 5 (alt.5), alto 6 (alt.6), tenor 1 (ten.1), tenor 2 (ten.2), tenor 3 (ten.3), tenor 4 (ten.4), bass 1 (bas.1), bass 2 (bas.2), bass 3 (bas.3), bass 4 (bas.4), and bass 5 (bas.5). The second staff includes the piano (pno), soprano (sopr), alto 1 (alt.1), alto 2 (alt.2), alto 3 (alt.3), alto 4 (alt.4), alto 5 (alt.5), alto 6 (alt.6), tenor 1 (ten.1), tenor 2 (ten.2), tenor 3 (ten.3), tenor 4 (ten.4), bass 1 (bas.1), bass 2 (bas.2), bass 3 (bas.3), bass 4 (bas.4), and bass 5 (bas.5). The third staff includes the piano (pno), soprano (sopr), alto 1 (alt.1), alto 2 (alt.2), alto 3 (alt.3), alto 4 (alt.4), alto 5 (alt.5), alto 6 (alt.6), tenor 1 (ten.1), tenor 2 (ten.2), tenor 3 (ten.3), tenor 4 (ten.4), bass 1 (bas.1), bass 2 (bas.2), bass 3 (bas.3), bass 4 (bas.4), and bass 5 (bas.5).

Measure 1: The piano (pno) plays eighth-note chords. The vocal parts sing eighth-note chords.

Measure 2: The piano (pno) plays eighth-note chords. The vocal parts sing eighth-note chords.

Measure 3: The piano (pno) plays eighth-note chords. The vocal parts sing eighth-note chords.

(47)   



4 

GUERRERO -18- Evans

(50)

**D**

$\text{♩} = 108$

4

4

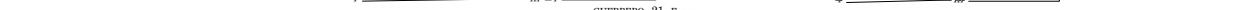
(53) 





GUERRERO -20- Evans

(56) 



(59) 





GUERRERO -22- Evans

62

4 4 4

62

63

64

⑤ 4

**E** = 90

85

86

GUERRERO -24- Evans

(88) 

CHAPTER 4. SCORE

GUERRERO -25- Evans

(71) 

The musical score consists of three staves of music for orchestra and piano. The first staff includes parts for soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, bass 5, and bass 6. The second staff includes parts for soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, bass 5, and bass 6. The third staff includes parts for soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, bass 5, and bass 6. The score is in common time, with a key signature of one sharp. Measure 1 starts with a dynamic of  $p$ . Measures 2 and 3 continue with the same instrumentation and dynamics.

(74) 

The musical score page 74 consists of three staves of music. The first staff includes parts for soprano (sopr), alto (alt), tenor (ten), bass (bas), and double bass (db). The second staff includes parts for soprano 1 (sopr.1), soprano 2 (sopr.2), alto 1 (alt.1), alto 2 (alt.2), tenor 1 (ten.1), tenor 2 (ten.2), tenor 3 (ten.3), tenor 4 (ten.4), bass 1 (bas.1), bass 2 (bas.2), and bass 3 (bas.3). The third staff includes parts for soprano 3 (sopr.3), soprano 4 (sopr.4), alto 3 (alt.3), alto 4 (alt.4), tenor 5 (ten.5), bass 4 (bas.4), and double bass (db). The music features various dynamics such as  $p$ ,  $f$ ,  $pp$ , and  $mf$ , and performance instructions like  $\text{f} \rightarrow \text{pp}$ . Measure numbers 24, 25, and 26 are indicated above the staves.

(77) 4 4 4

A page of a musical score for orchestra, page 80, showing measures 4 through 7. The score includes parts for soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, and cello. The music features complex rhythmic patterns and dynamic markings like f, pp, and mf.

(83)

(86) 
 The musical score consists of three staves of music for orchestra and piano. The first staff includes parts for soprano, alto, tenor, bass, and piano. The second staff includes parts for soprano, alto, tenor, bass, and piano. The third staff includes parts for soprano, alto, tenor, bass, and piano. The score is in common time, with measure numbers 1, 2, and 3 indicated above each staff. Measure 1 starts with a forte dynamic. Measures 2 and 3 show a continuation of the musical line with various dynamics and articulations.

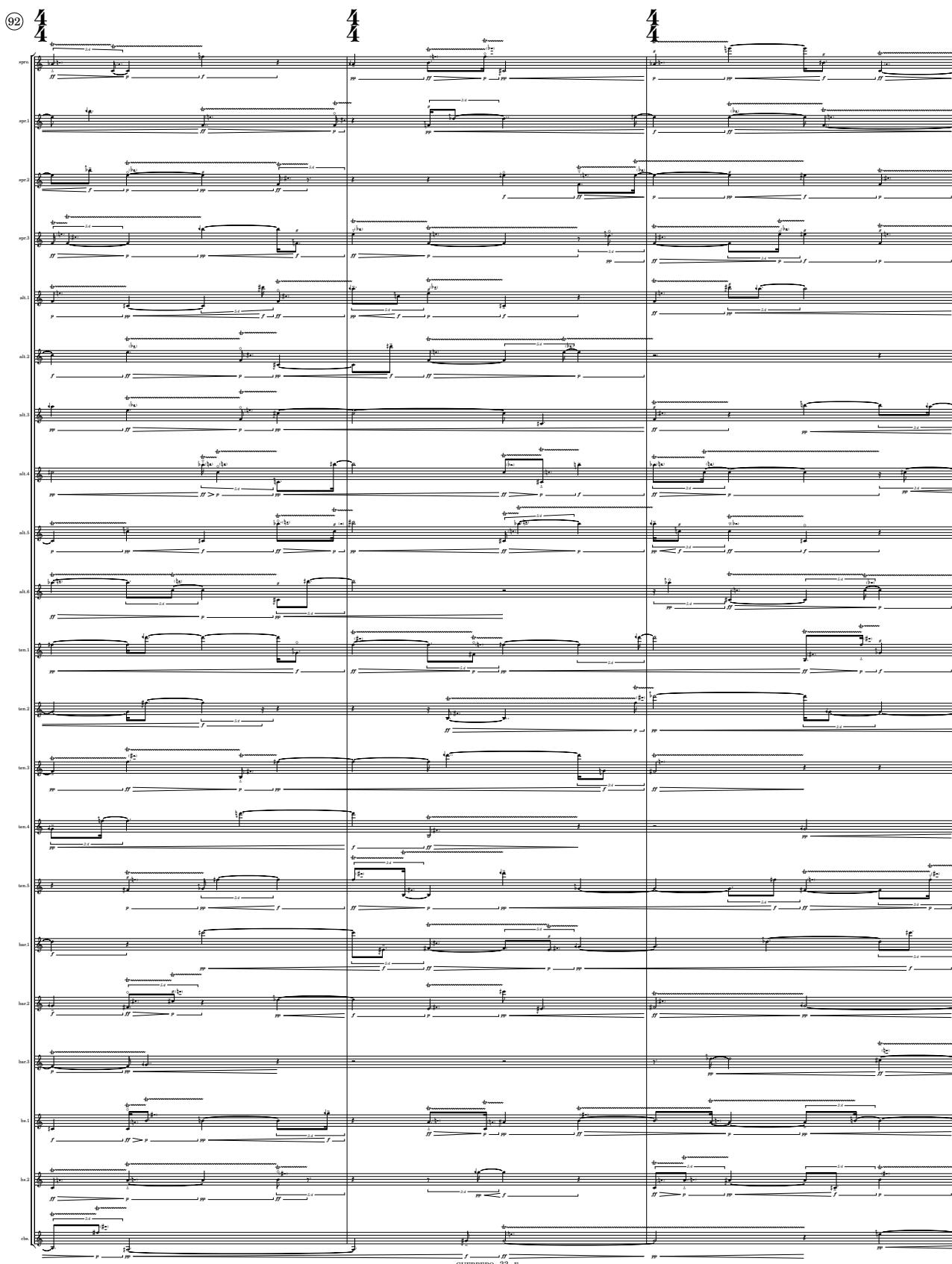
(89)

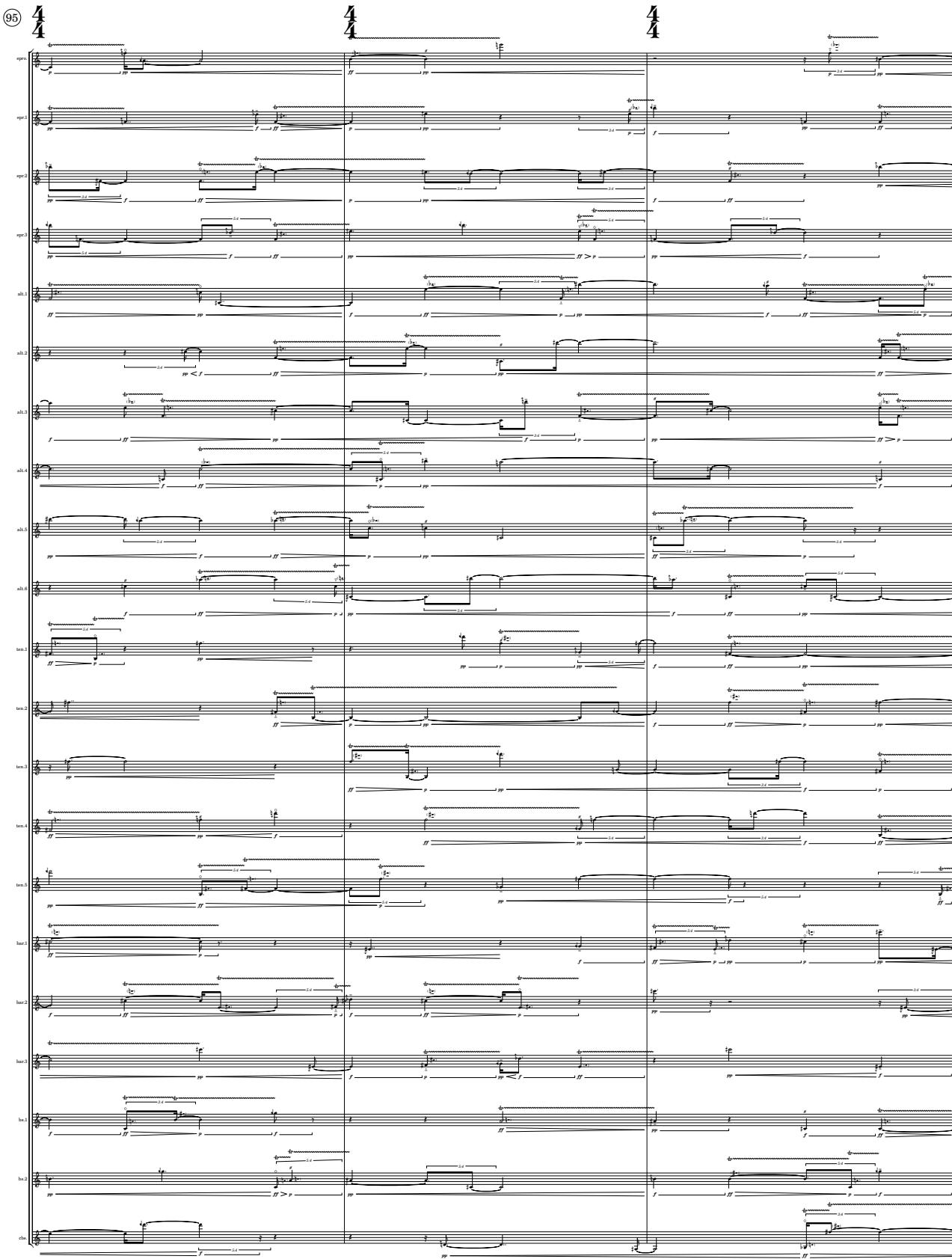
**A**

**B**

**C** [F]

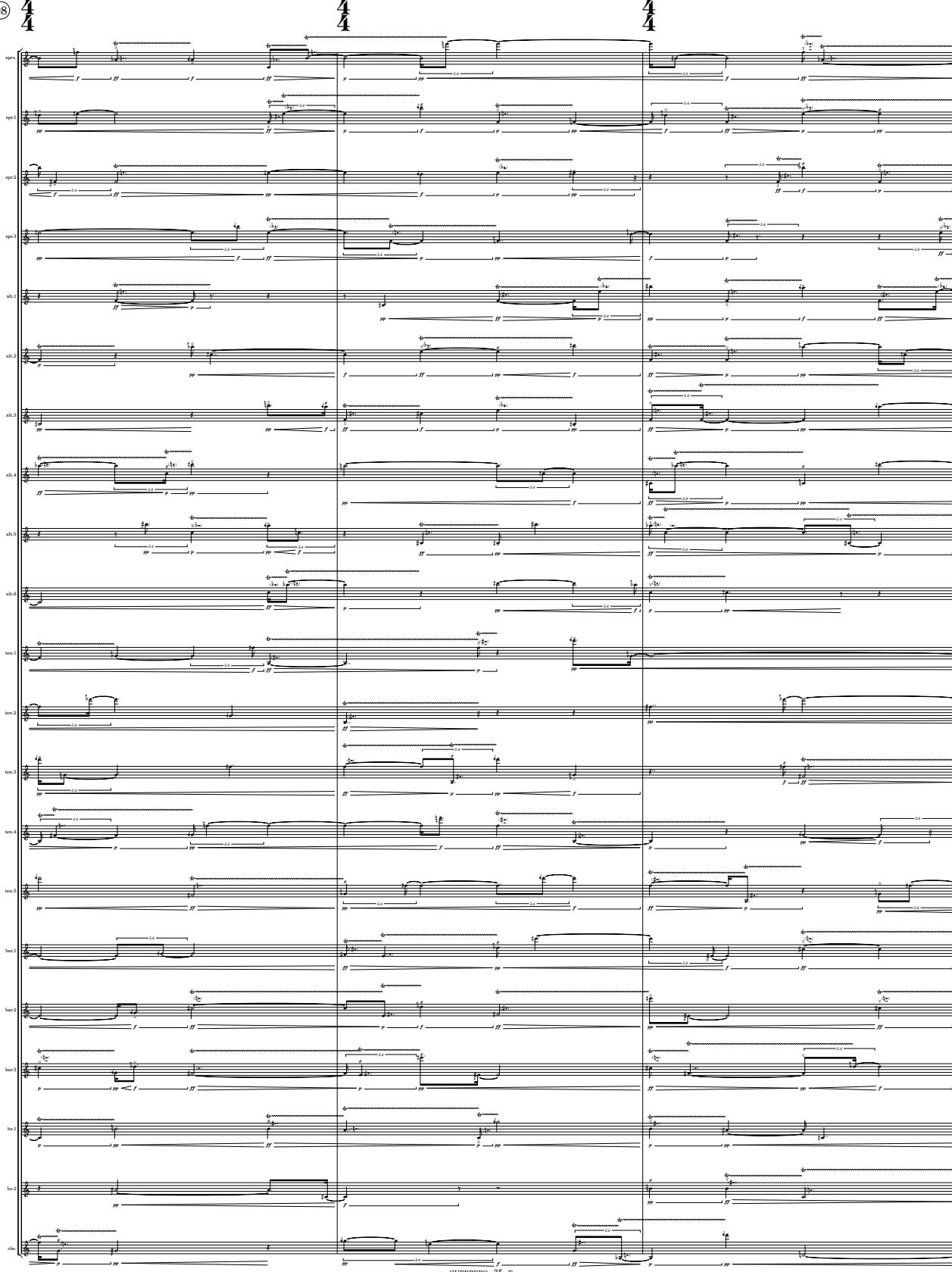
GUERRERO -32- Evans

(92) 
 The musical score consists of three staves of music for orchestra and piano. The first staff includes parts for soprano, alto, tenor, bass, and bassoon. The second staff includes parts for soprano, alto, tenor, bass, and bassoon. The third staff includes parts for soprano, alto, tenor, bass, and bassoon. The piano part is located at the bottom of each staff. The music is in common time (indicated by a '4'). Measure 1 starts with a dynamic of  $\text{f}$ . Measures 2 and 3 continue with various dynamics including  $\text{pp}$ ,  $\text{p}$ ,  $\text{f}$ , and  $\text{ff}$ . The vocal parts show sustained notes and rhythmic patterns typical of choral music.

(95) 





(98) 

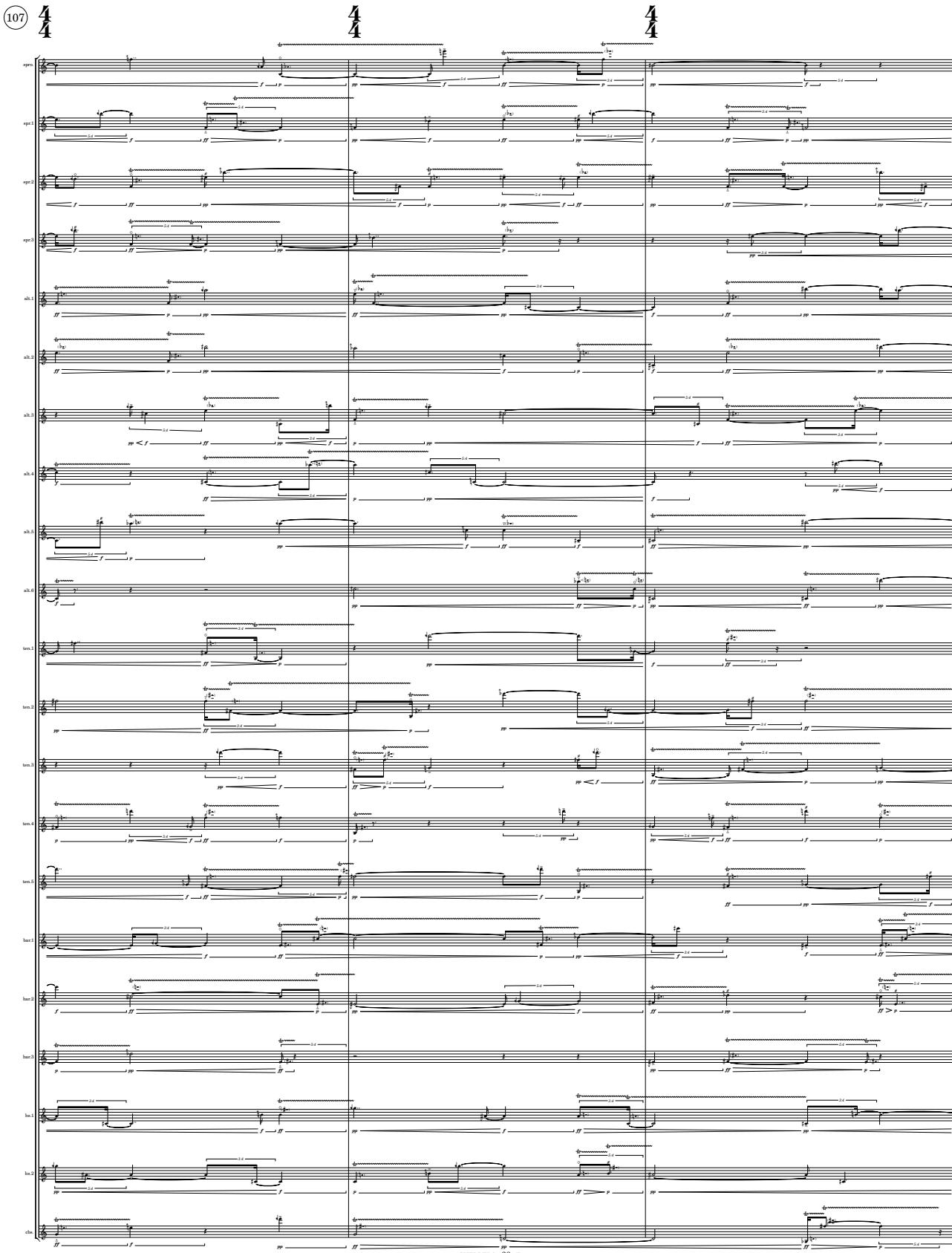
GUERRERO -35- Evans

(101)

GUERRERO -36- Evans

(104)

GUERRERO -37~ Evans

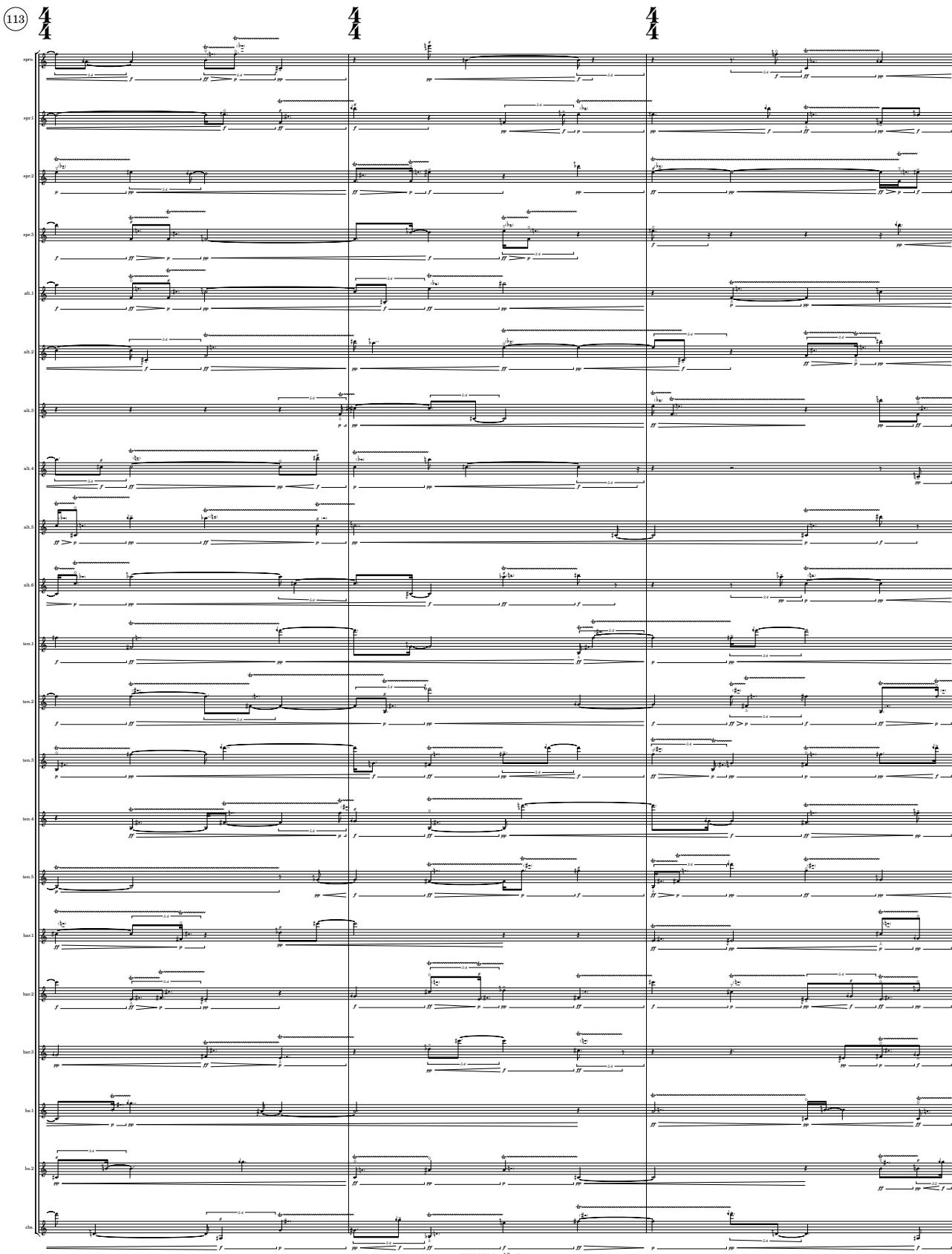
(107) 

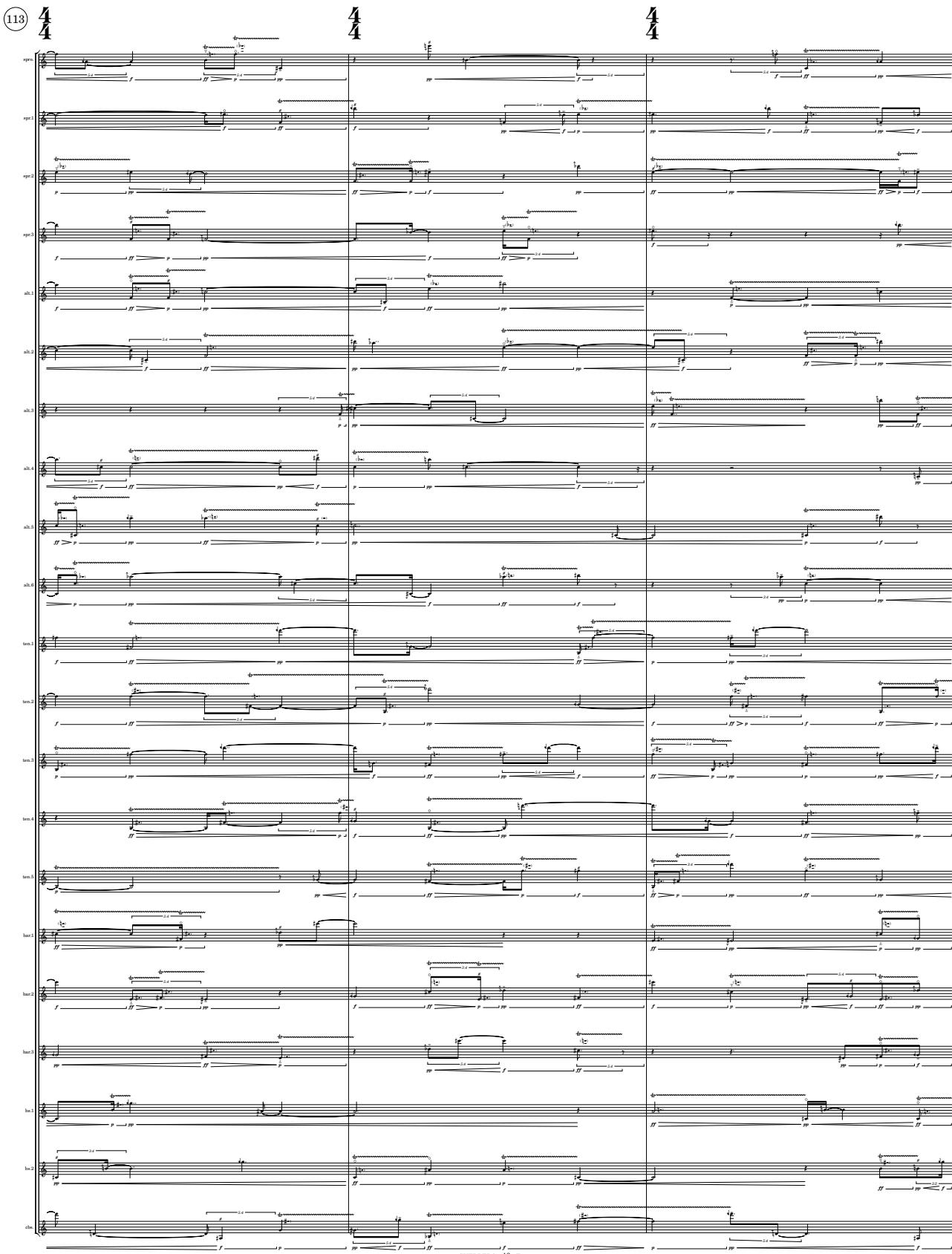
**GUERRERO -38- Evans**

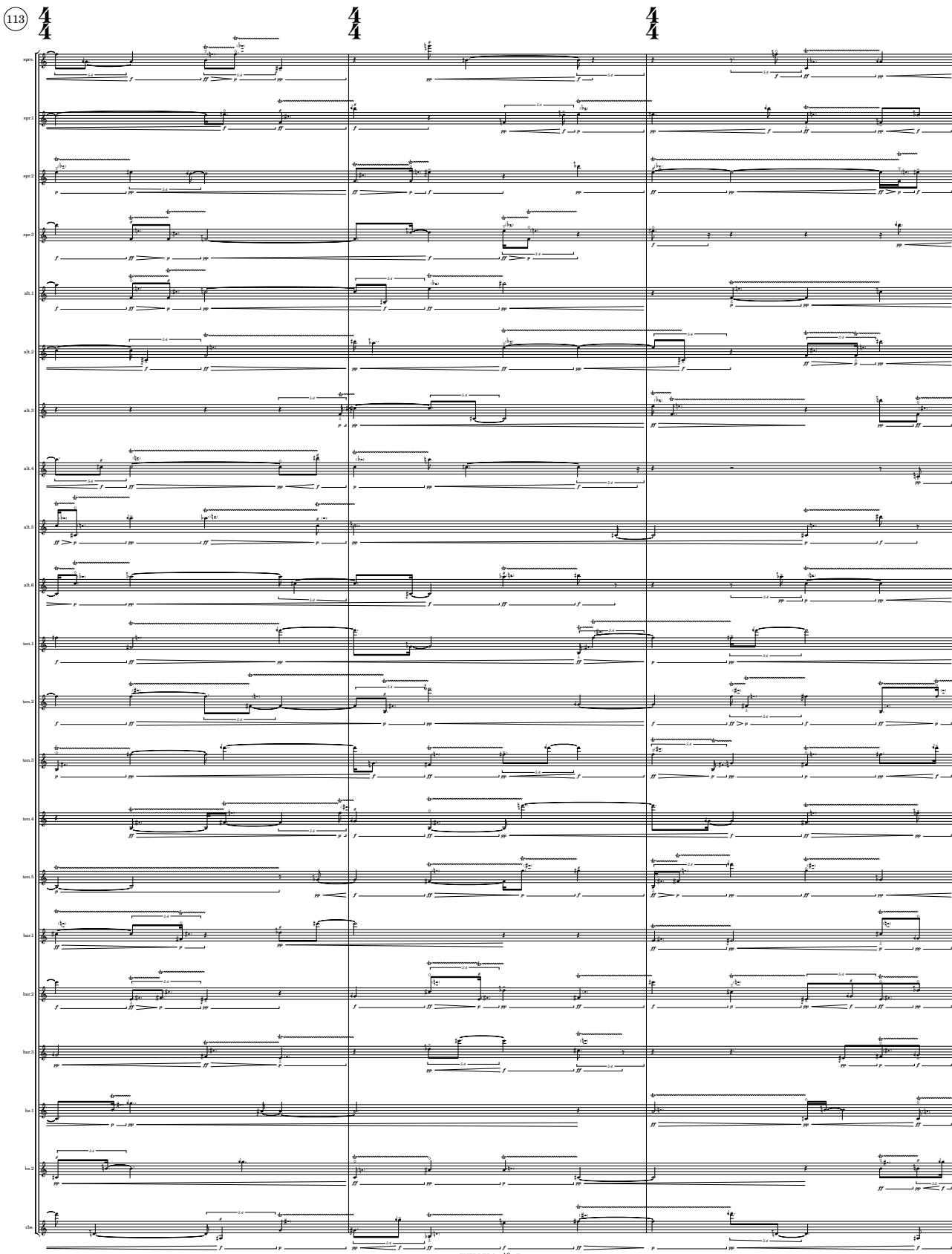
110

4

4

(113) 



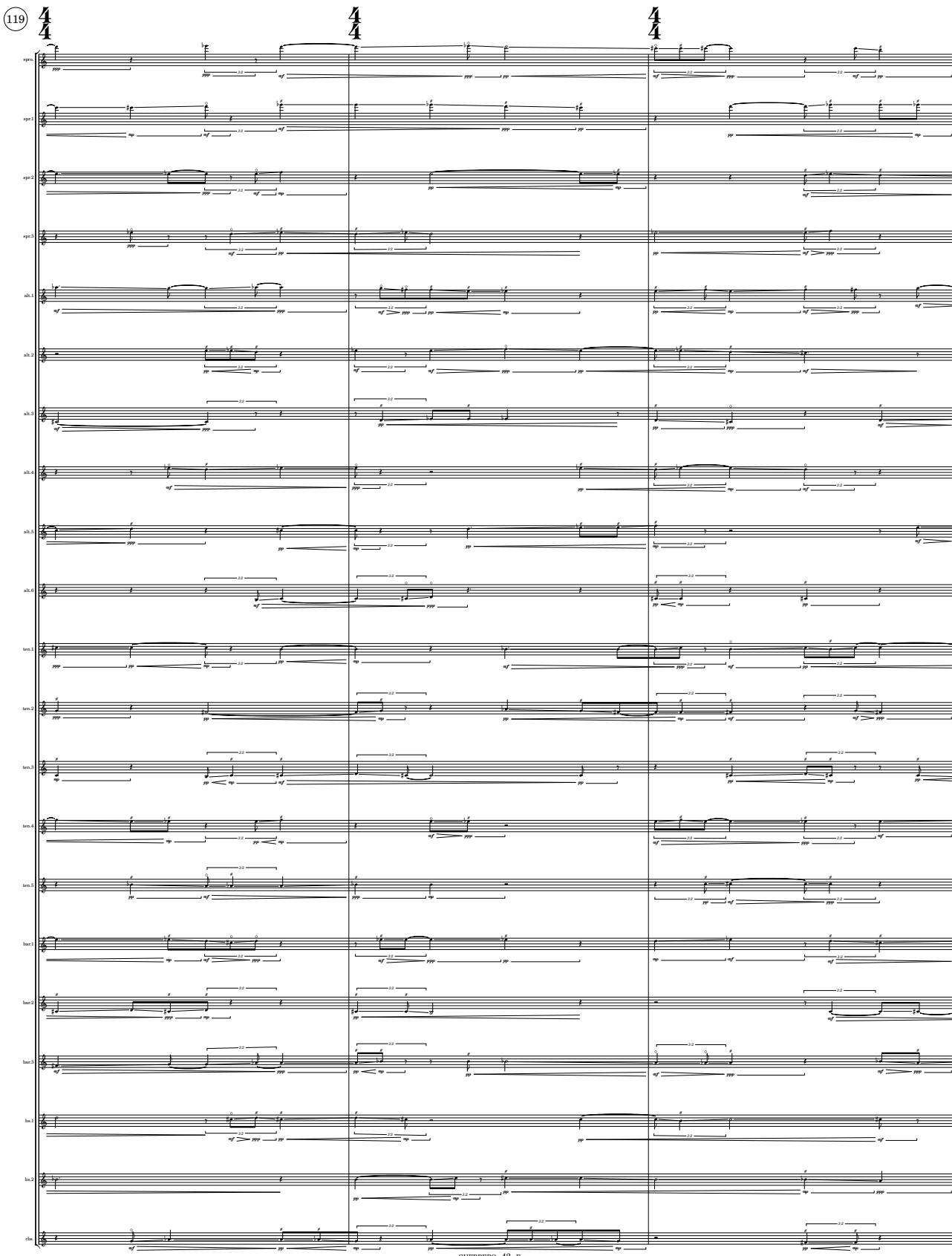


(116)

The musical score page contains three staves of music. The first staff includes parts for soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, and bass 5. The second staff includes parts for soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, and bass 5. The third staff includes parts for soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, and bass 5.

Instrumental parts listed on the left side of the page include: soprano 1, soprano 2, soprano 3, alto 1, alto 2, alto 3, alto 4, alto 5, alto 6, tenor 1, tenor 2, tenor 3, tenor 4, tenor 5, bass 1, bass 2, bass 3, bass 4, and bass 5.

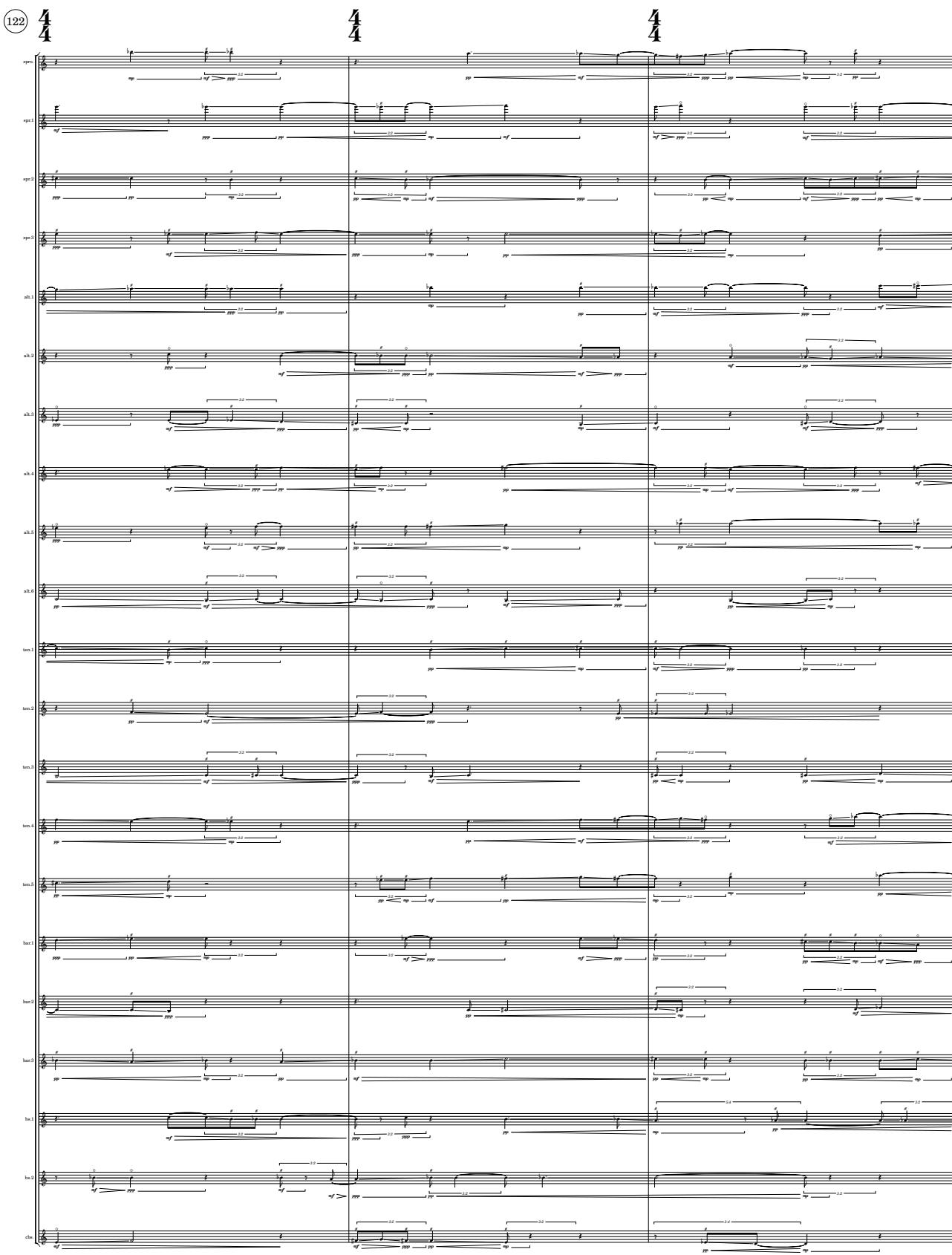
At the bottom of the page, the text "GUERRERO -41- Evans" is visible.

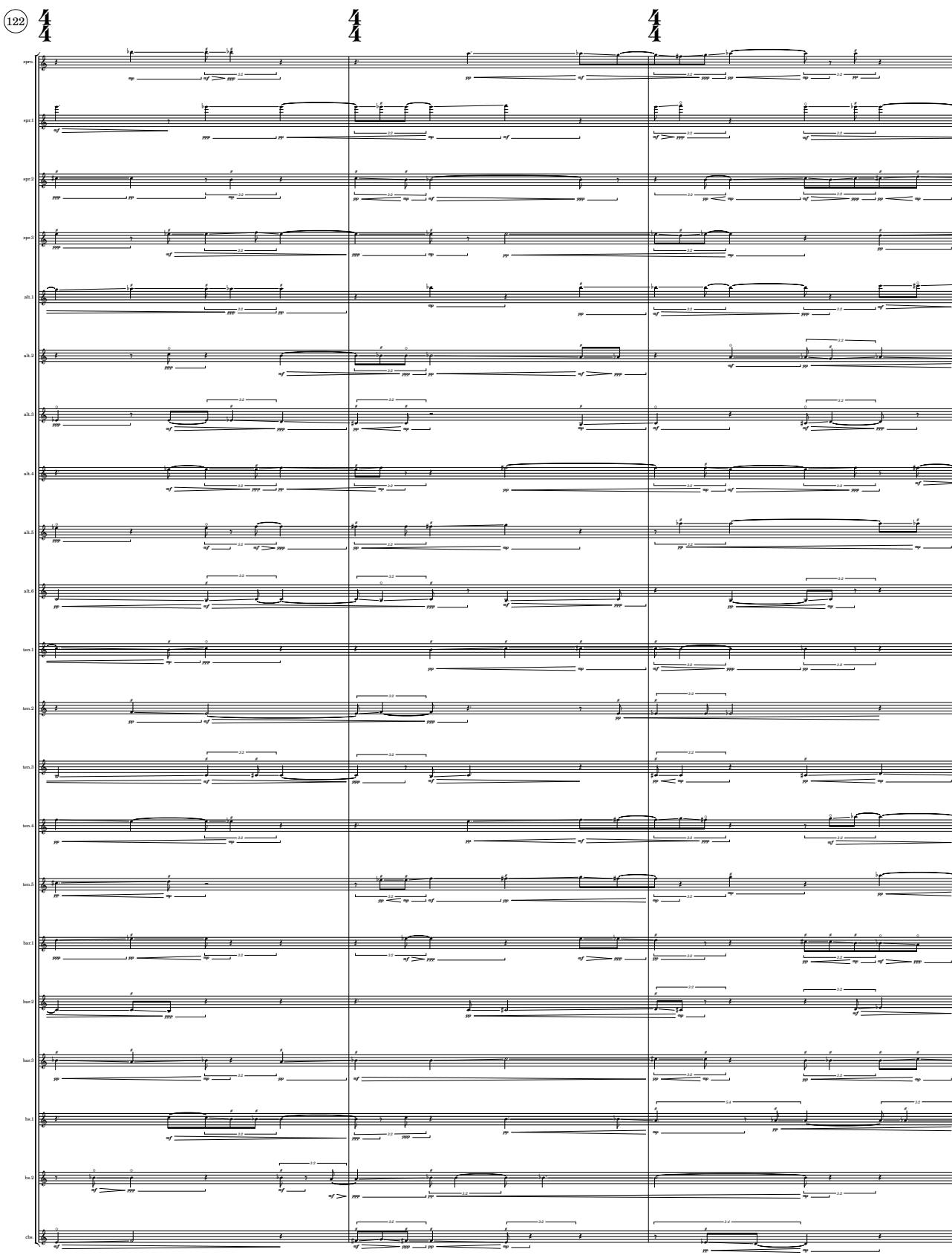
(119) 

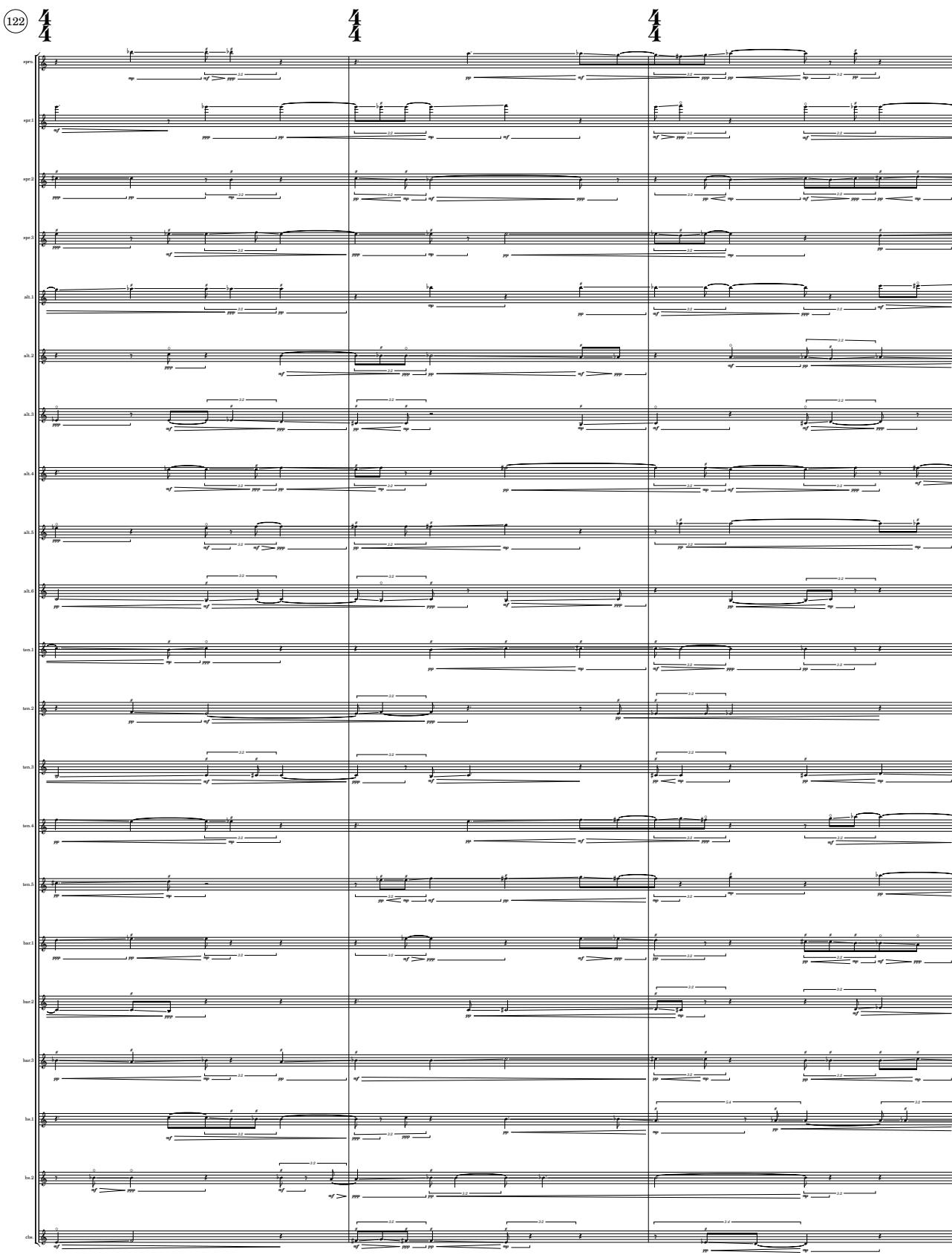
119

sopr.  
alt.  
ten.  
bas.  
tuba

GUERRERO -42- Evans

(122) 





GUERRERO -43- Evans

(125)

GUERRERO -44- Evans

(128)

128 60

129

130

sopr.1  
sopr.2  
sopr.3  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
alt.6  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bar.1  
bar.2  
bar.3  
bass.1  
bass.2  
bass.3  
tuba

GUERRERO -45- Evans

(131)

**2**

**4**

**3**

GUERRERO -46- Evans

(134)

**3**

**4**

**5**

sopr.

alt.1

alt.2

alt.3

alt.4

alt.5

ten.1

ten.2

ten.3

ten.4

ten.5

bass.1

bass.2

bass.3

bsn.

measures 134-135

measures 135-136

measures 136-137

GUERRERO -47~ Evans

(137)

**3**

**4**

**5**

spn.1  
spn.2  
spn.3  
alt.1  
alt.2  
alt.3  
alt.4  
alt.5  
alt.6  
ten.1  
ten.2  
ten.3  
ten.4  
ten.5  
bsn.1  
bsn.2  
bsn.3

GUERRERO -48- Evans

140

**4**

**4**

sopr.

alto

ten.

bassoon

harp

double bass

cello

GUERRERO -49- Evans

(143) 





GUERRERO -50- Evans

(146)

149

4

4

4

sopr.

apr.1

apr.2

apr.3

alt.1

alt.2

alt.3

alt.4

alt.5

alt.6

ten.1

ten.2

ten.3

ten.4

ten.5

bass.1

bass.2

bass.3

bass.4

cbs.

(152)

**A**

**B**

**C**

GUERRERO - 53 - Evans

155

4 4 4

spn.  
spz.1  
spz.2  
spz.3  
ah.1  
ah.2  
ah.3  
ah.4  
ah.5  
ah.6  
tm.1  
tm.2  
tm.3  
tm.4  
tm.5  
bar.1  
bar.2  
bar.3  
bar.4  
bar.5  
chz.

(158) **4**

**8**

GUERRERO -55- Evans

Other scores from Gregory Rowland Evans include:

**UNACCOMPANIED**  
*Five Excuses* (for cello alone)  
*Five Excuses* (for piano alone)  
*Epiphora* (for solo cello)  
*Five Excuses* (for xiao alone)  
*soNOTina* (for solo piano)

**CHAMBER**  
*String Trio no.1*  
*Violin Concerto*  
*Five Excuses* (for string trio)  
*Adumbration* "String Trio 2"  
*Hamon shi* "String Quartet 1"  
*Cthar* (for two cellos)  
*Four Ages of Sand* (fl asax vc)

**ELECTROACOUSTIC**  
*Bewegt die Erde*  
*D.E.i. : Doppelgänger* (cello, violin)  
*D.E.vii. : Stach* (for laptop ensemble)  
*D.E.i. : NGC 3270* (for percussion trio)  
*D.E.ii. : Cernunnosel* (for viola)  
*D.E.iv. : Arboralkarstographie* (for cello)  
*Siderus Nuncius* (for oboe)

**ORCHESTRAL**  
*Arquitectura 11611*  
*Metamorphoses* (after Illouz)  
*Tianshi* (for 12 players)  
*GUERRERO* (21 saxophones)



# Bibliography

- [1] Bača, Trevor et al., *Abjad: An Open-Source Software System for Formalized Score Control*, abjad.mbrsi.org, 2015
- [2] Besada, Joseé L., *Metamodels in Compositional Practices*, Editions Delatour, 2017
- [3] Burkholder, J. Peter et al., *A History of Western Music 9th ed.*, W.W. Norton & Company, 2014
- [4] Carter, Elliott, *Harmony Book*, Carl Fischer, 2002
- [5] Forte, Alan, *The Structure of Atonal Music*, Yale University Press, 1977
- [6] Guerrero, Francisco, *Rhea*, Edizioni Suvini Zerboni, 1988
- [7] Lee, Shun-Yang, *Time-Frequency Analysis of the Shepard Tone*, Bard College, 2010
- [8] Weiss, Marcus et al., *Die Spieltechnik des Saxophons*, Bärenreiter, 2010