

Guilherme Rafael Soares

Luteria Composicional de algoritmos pós-tonais

10 de julho de 2014, v0.6-Qualificação

Guilherme Rafael Soares

Luteria Composicional de algoritmos pós-tonais

Prévia da dissertação para a banca de qualificação para o Mestrado em Arte, Cultura e Linguagens do IAD-UFJF.

UFJF - Universidade Federal de Juiz de Fora

Instituto de Artes e Design

Programa de Pós-Graduação em Artes, Cultura e Linguagens

Orientador: Prof. Dr. Daniel Quaranta

10 de julho de 2014, v0.6-Qualificação

Guilherme Rafael Soares

Luteria Composicional de algoritmos pós-tonais / Guilherme Rafael Soares. – ,
10 de julho de 2014, v0.6-Qualificação-
55 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Daniel Quaranta

Tese (Mestrado) – UFJF - Universidade Federal de Juiz de Fora

Instituto de Artes e Design

Programa de Pós-Graduação em Artes, Cultura e Linguagens, 10 de julho de 2014,
v0.6-Qualificação.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador: Prof. Dr. Daniel Quaranta
II. UFJF - Universidade Federal de Juiz de Fora. III. Instituto de Artes e Design
IV. Luteria Composicional de algoritmos pós-tonais

CDU 02:141:005.7

Guilherme Rafael Soares

Luteria Composicional de algoritmos pós-tonais

Prévia da dissertação para a banca de qualificação para o Mestrado em Arte, Cultura e Linguagens do IAD-UFJF.

Trabalho aprovado , 13 de fevereiro de 2015:

Prof. Dr. Daniel Quaranta
Orientador

Professor
Convidado 1

Professor
Convidado 2

10 de julho de 2014, v0.6-Qualificação

Resumo

Esta pesquisa visa problematizar e sistematizar um catálogo de experimentos constituído de pequenas peças musicais e seus algoritmos geradores, objetivando a construção de uma biblioteca de objetos para composição assistida por computador que gere partituras baseadas em regras quantitativas extraídas de análises musicais.

Formalizamos tais aspectos através de um estudo comparado de dois paradigmas de análise musical: "*A Teoria Gerativa da Música Tonal*" (LERDAHL; JACKENDOFF, 1983) com algumas de suas continuidades (LERDAHL, 2009; TEMPERLEY, 2001) e a "*Teoria de grupos das classes de alturas*" (ou "*Pitch Class Set Theory*") (FORTE, 1973; STRAUS, 2004).

Os procedimentos são demonstrados a partir de aspectos singulares de algumas peças da suíte Mikrokosmos do compositor Béla Bartók, gerando composições algorítmicas a partir das regras observadas. Este repertório foi escolhido devido a seu reconhecido contexto como composições pianísticas e pedagógicas situadas nas fronteiras da pós-tonalidade.

Apontamos as limitações encontradas na aplicação dos paradigmas analíticos adotados aqui no contexto da suíte de peças escolhidas e suas derivações composicionais.

Detalhamos questões computacionais para esta implementação e deixamos um legado de código aberto para continuidades possíveis deste trabalho.

Palavras-chaves: Música algorítmica. Pós-tonalismo. Teoria dos conjuntos. Pitch class theory. Luteria. Composição assistida por computador. Cibernética. Software livre. Cognição musical. Teoria Gerativa da Música Tonal. Mikrokosmos. Arte Sonora.

Lista de ilustrações

Figura 1 – Objeto Chord-Seq do OM	29
Figura 2 – Operações com listas no OMs	30
Figura 3 – Operações com listas no OMs	31
Figura 4 – Alguns objetos para operações combinatórias e operações em grupos de classes de altura no OMs	32
Figura 5 – Segmentação em Chord-Seq de OpenMusic	32
Figura 6 – Manipulação de arquivos MIDI no PD.	34
Figura 7 – Objetos da biblioteca "Real Time Composition" para PD.	35
Figura 8 – Objetos da biblioteca "list-abs"para PD.	36
Figura 9 – Objeto [probalizer].	37
Figura 10 – Gerador de um acorde Dó maior (dó4 e4 g5) na clave de sol em Lilypond	41
Figura 11 – Gerador de uma nota dó4 na clave de sol em MusicXML	42

Lista de abreviaturas e siglas

GTMM	<i>Generative Theory of Tonal Music</i> ¹
TPS	<i>Tonal Pitch Space</i> ²
CBMS	<i>Cognition of Basic Musical Structures</i> ³
OM	<i>Open Music</i> ⁴
PD	<i>Pure Data</i> ⁵

¹ "Teoria Gerativa da Música Tonal"(LERDAHL; JACKENDOFF, 1983)

² "Espaço das Alturas Tonais"(LERDAHL, 1988)

³ "Cognição das Estruturas Musicais Básicas"(TEMPERLEY, 2001)

⁴ <<http://repmus.ircam.fr/openmusic/home>>. Acessado em 10 de julho de 2014.

⁵ <<http://puredata.info>>. Acessado em 10 de julho de 2014.

Sumário

	Introdução	13
I	MUSICA GENERATIVA E O "READY-MADE"POS-TONAL	17
1	CATALOGO DE ALGORITMOS IDIOSSINCRÁTICOS EM BELA BARTOK	21
1.1	Contextos analíticos e singularidade estrutural em Bela Bartok . . .	21
1.2	Idiosincrasia dos Ciclos Intervalares	21
1.2.1	Eixo de Simetrias	22
1.2.2	Celula Intervalar	22
1.2.3	Harmonização de melodias folclóricas	22
1.2.4	Centricidade Tonal	22
1.3	Teses sobre a Secção Aurea	22
1.4	Algoritmos Ready-Made na suite Mikrokosmos	22
1.4.1	Contraponto Modal	22
1.4.2	Polimodalismo	22
1.4.3	Rítmica	22
II	IMPLEMENTAÇÃO COMPUTACIONAL	23
2	ANÁLISE ASSISTIDA POR COMPUTADOR	25
3	COMPOSIÇÃO ASSISTIDA POR COMPUTADOR	27
3.1	Linguagens Dataflow	27
3.1.1	OM	28
3.1.1.1	chord-seq	28
3.1.1.2	Listas de dados	30
3.1.1.3	Operações combinatórias e operações sobre grupos de classes de altura	31
3.1.2	PD	33
3.1.2.1	Biblioteca [RTC-lib]	33
3.1.2.2	Biblioteca de manipulação de listas [list-abs]	34
3.1.2.3	objeto [probalizer]	34

4	ARQUIVOS E SCRIPTS PARA SEGMENTAÇÃO DE DADOS MUSICAIS	39
4.1	MIDI	39
4.2	Lilypond	41
4.3	MusicXML	41
4.4	Bibliotecas Python auxiliares	42
4.4.1	Abjad	42
4.4.2	Music 21	43
III	EXPERIMENTOS COMPOSICIONAIS GENERATIVOS	45
5	GERADOR DE MIKROBAGATELLAS	47
6	COSMOSONORIDADE DE UM RASCUNHO EM ABERTO	49
	Conclusão	51
	Referências	53

Introdução

Desde o momento em que o computador emancipa-se do estúdio experimental e seus aparelhos caros e institucionais e possibilita o processamento de dados em tempo real em gadgets que cabem no nosso bolso (e cada vez mais até dentro dos nossos corpos) fala-se constantemente na possibilidade de interação com a transformação de dados audiovisuais através de uma computabilidade da escritura composicional ou do gestual performático.

Em seu livro sobre mediação tecnológica contemporânea na composição Fernando Iazzetta (2009) fala sobre um tipo de *"luteria composicional"* que surge do experimento de estúdio migrando para os computadores pessoais, onde a criação dos instrumentos (que na verdade são códigos, procedimentos computacionais, "patches") agora já fariam parte do processo composicional:

“Mesmo porque, muitas vezes, o trabalho de composição se confunde com o trabalho de criação dos instrumentos que serão usados na composição. O conhecimento do funcionamento interno destes instrumentos e a possibilidade de correção e aperfeiçoamento constante assim como o acoplamento de novas interfaces ao sistema, confere ao compositor um domínio maior da execução da sua obra” (IAZZETTA, 2009, p. 209).

Por outro lado, o fechamento deste processo em *"microteorias composicionais derivadas da circulação dos manuais de softwares musicais"* (IAZZETTA, 2009, p. 152) não parecem serem suficientes para dar conta de uma série de procedimentos composicionais que existiam muito antes de serem pensados a priori já por dentro destes sistemas.

"Qualquer estrutura, gramática ou modelo pode, em princípio, ser transposto para o âmbito sonoro com a intenção de produzir música. Uma vez que nos sistemas computacionais todo e qualquer elemento é transcrito na forma de símbolos abstratos do mesmo tipo (em última instância, bits representados por 0 e 1), esse tipo de procedimento se torna tentador, mas também vulnerável.(...)Certamente estas transposições de um campo a outro não destroem a coerência interna dos fenômenos transpostos, mas de forma alguma asseguram a geração de uma coerência musical, pelo menos não no nível perceptivo. (...) **O discurso enfatizando o caráter inovador que acompanha cada novo invento geralmente esconde o quanto nossos avanços representam uma consolidação de conhecimentos existentes, mais do que saltos progressivos**". (IAZZETTA, 2009, p. 151-153, grifo nosso.)

Esta pesquisa propõe um recorte específico de alguns procedimentos composicionais emergentes na primeira metade do século XX, que estão no limite entre o politonalismo e o atonalismo, sob a luz de teorias que influenciaram a criação de algoritmos de análise musical assistida por computador nas décadas mais recentes.

Os problemas computacionais considerados no percurso compõem uma suíte de objetos e funções organizados em bibliotecas para as linguagens de programação musical OpenMusic e Puredata, facilita-se assim um estudo comparado das implementações dos procedimentos algorítmicos em diferentes sintaxes. As composições geradas pelo processo fomentam nova reflexão sobre o automatismo e interação dentro de seus procedimentos.

Complementamos o trabalho com a documentação de *scripts* Python auxiliares para formatação e segmentação de partituras (em formatos midi, musicxml ou lilypond, dependendo do caso).

O percurso deste trabalho se dá em duas etapas: **Parte I:Paradigmas para uma Análise Musical Pós-Tonal** e **Parte II:Implementação Computacional**.

Na **Parte I** buscamos organizar bases para análises computacionais do processo criativo traçando uma epistemologia das gramáticas musicais a partir da influência que a linguística teve na musicologia. Atentamos para as teorias derivadas da pesquisa de [Chomsky \(1957\)](#) e sua aplicação no processamento de linguagens naturais. Buscamos aspectos que direcionaram pesquisas musicológicas para a possibilidade de aplicar regras analíticas em sistemas computáveis.

A partir dessas perspectivas, encontramos duas abordagens analíticas que nos chamaram atenção por serem relativamente contemporâneas entre si e que partem de diferentes princípios.

A primeira abordagem é a da "*Teoria Gerativa da Música Tonal*" ([LERDAHL; JACKENDOFF, 1983](#)) e alguns desdobramentos mais recentes como a "*Cognição das Estruturas Musicais Básicas*" ([TEMPERLEY, 2001](#)) para buscar algoritmos que definam critérios quantitativos para análises tomem em consideração a normatização operada pela música tonal ocidental na percepção do seu ouvinte médio. Estas teorias flertam com argumentos da psicologia cognitivista da música ([KRUMHANS, 1990](#)) para justificar seus pressupostos.

A segunda é a corrente musicológica que trabalha sobre um mapeamento dos agrupamentos de classes de alturas, organizando taxonomias ([FORTE, 1973](#)), descrevendo operações de transformação([STRAUS, 2004](#)) e buscando critérios mais autorais para apontar singularidades em composições ([LESTER, 1989](#); [STRAUS, 2004](#)) que não necessariamente operam sobre os pressupostos de uma expectativa de tonalidade explícita.

Na **Parte II** trabalhamos uma reflexão sobre procedimentos, formatos de arquivos e bibliotecas de linguagem de programação para uma "*Composição Assistida por Computador*"(CAC) usadas durante esta pesquisa. Bases para implementação dos estudos da **Parte I**.

E finalmente na conclusão⁶ deste trabalho utilizaremos algumas ideias e particularidades retiradas dos procedimentos analíticos detalhados na [Parte I](#) para construir composições musicais que aplicam estruturas destacadas das análises de *Mikrokosmos* de Béla Bartók, usando as implementações computacionais que nos pareceram mais estáveis e bem documentadas.

⁶ ver [Capítulo 6](#)

Parte I

Musica Generativa e o "ready-made" pos-tonal

"Found Mathematical Objects"([JOHNSON, 2001](#))

mikrokosmos 136 ([ROIG-FRANCOLÍ, 2008](#), p. 49) apud ([FILHO, 2012](#))

1 Catálogo de algoritmos idiossincráticos em Bela Bartók

1.1 Palavras Chave

"mapear-se em"

"Simetria inversiva" ([STRAUS, 2004](#), pg.86) mikrokosmos 109 Bali mikrokosmos 101 Diminished Fifth

"Simetria Transpositiva"

"Centricidade"

1.2 Contextos analíticos e singularidade estrutural em Bela Bartók

1.3 Idiossincrasia dos Ciclos Intervalares

([MORGAN; ANTOKOLETZ, 1991](#))

octatonica

pitch class sets ([COHN, 1991](#))

pitch centrality ([COHN, 1991](#), p. 4-20)

serie harmônica ([LENDVAI, 1962](#)) citero2013acusticabartok

rotacional ([KELLER, 2011](#))

Ever since the publication of Edwin von der Nüll's (1905–1945) Béla Bartók. Ein Beitrag zur Morphologie der neuen Musik (1930), the phenomenon of note structures that contain major and minor thirds simultaneously has been a well-known and frequently discussed topic in research on Bartók; e.g., the major-minor chord, set 0,3,4,7. Nüll (ibid.: 74) deemed the major-minor chord a neutral sound, claiming that simultaneous sounding of the two thirds produces an "absence of mode" (Geschlechtslosigkeit). 23 Nüll (ibid.: 73) sees Bartók as an evolutionary phenomenon in matters of form and pitch organization alike, in contrast to Arnold Schoenberg (1874–1951), invariably starts out from precedent and goes on from there to unfold further possibilities. In *Moderne Harmonik*, as regards Bartók's pitch organization, Nüll (1932) asserts that it is tertian, tonal, and diatonic.

I must state that all my music is determined by instinct and sensibility; no one need ask me why I wrote this or that or did something in this rather than in that way. I

could not give any explanation other than I felt this way, or I wrote it down this way. I never created new theories in advance. I hated such ideas. I had of course, a very definite feeling about certain directions to take, but at the time of the work, I did not care about the designations, which would apply to those directions or to their sources. This attitude does not mean that I composed without set plans and without sufficient control. The plans were concerned with the spirit of the new work and with technical problems (for instance formal structure involved by the spirit of the work) all more or less instinctively felt, but I never was concerned with general theories to be applied to the works I was going to write. Now that the greatest part of my work has already been written, certain general tendencies appear – general formulas, which theories can be deduced. (B Bartok Essays BE 1976 [1943]: 376.)

1.3.1 Eixo de Simetrias

(LENDVAI; BUSH, 1971)

1.3.2 Celula Intervalar

1.3.3 Harmonização de melodias folclóricas

1.3.4 Centricidade Tonal

1.4 Teses sobre a Secção Aurea

(BACHMANN; BACHMANN, 1979)

Fibonacci 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

$\varphi = 0.618...$

1.5 Algoritmos Ready-Made na suite Mikrokosmos

1.5.1 Contraponto Modal

1.5.2 Polimodalismo

1.5.3 Rítmica

Parte II

Implementação Computacional

2 Análise Assistida por Computador

<http://web.mit.edu/music21/doc/moduleReference/moduleSearch.html>

3 Composição Assistida por Computador

O objeto central de estudo desta pesquisa são os algoritmos de manipulação de dados de uma superfície musical normatizada para a escala de temperamento igual, de doze tons ou "12-TET" (SETHARES, 2005, p. 76), tradicionalmente usada nas partituras da prática comum. Estamos abstraindo no momento as questões como timbre, microtonalismo, afinações ou quaisquer outros aspectos não formalizáveis em partitura tradicional.

Curiosamente, a chamada "computer music" (ROADS, 1996) que derivou dos primeiros mainframes e seus métodos estocásticos de compositores como Hiller e Xenakis, parece cada vez mais seduzir para um estudo da música que inicia pela morfologia do espectro sonoro, onde impera o "microsom" (ROADS, 2004) manipulado nos níveis limites da psicoacústica. Interessa em nossa abordagem repensar um pouco os meandros desta manipulação simbólica de escalas e aglomerados de alturas, antes de mergulhar nas possibilidades expandidas.

O criador da linguagem PureData, categoriza esta diferença no prefácio do *"Open Music Composer's Book 1"* (AGON, 2006):

O campo da *computer music* pode ser pensado como tendo dois ramos fundamentais, um preocupado com a manipulação de sons musicais, e outra preocupada com a com a representação simbólica da música. (...) Os dois ramos podem provisoriamente serem chamados pelos nomes "Musica Gerada Por Computador" (o termo de Denis Baggi) e "Composição assistida por computador- ou pelos acrônimos MGC e CAC." (PUCKETTE, 2006, pg. ix)¹

3.1 Linguagens Dataflow

As linguagens de programação PureData (ou PD) e OpenMusic (ou OM) têm ao menos duas coisas em comum: ambas utilizam o paradigma de programação *dataflow* – uma representação gráfica dos algoritmos que deixa os programas similares a caixas conectadas por cabos, estimulando a imaginação para algo mais tátil do que cálculos abstratos.

Ambas também são linguagens que emergem a partir de projetos incubados no IRCAM (*Institut de Recherche et Coordination Acoustique/Musique*) a instituição francesa que tem entre seus idealizadores o compositor Pierre Boulez e é pioneira em pesquisas

¹ The field of computer music can be thought of as having two fundamental branches, one concerned with the manipulation of musical sounds, and the other concerned with symbolic representations of music. (...) The two branches might provisionally be given the names "Computer Generated Music" (Denis Baggi's term for it) and "Computer Aided Composition"— or CGM and CAC for short. (PUCKETTE, 2006, p. ix)

computacionais guiadas por processos composicionais. São descendentes diretas da primeira geração de linguagens musicais *dataflow*: *Patchwork* (no caso do OM) e *Max* (no caso do PD).

Considerando que são ainda muito utilizadas por pesquisadores de Composição Assistida por Computador (CAC) e Música Algorítmica, estas já podem de alguma forma ser consideradas linguagens de computação musical com relevância histórica o suficiente para no mínimo fundamentarem novas invenções.

Utilizamos nesta pesquisa ambas as linguagens testando suas limitações e disponibilidade de bibliotecas já prontas e bem documentadas.

3.1.1 OM

"Enquanto a maioria das “linguagens de programação musical” lidam principalmente com processamento de sinal e síntese sonora, uma abordagem original adotada pelo time de representação musical do IRCAM no fim dos anos 80 foi particularmente um foco na nas estruturas simbólicas e processos musicais, isto é, aspectos tradicionalmente ignorados ou deixados de lado dos ambientes computacionais." (BRESSION; AGON; ASSAYAG, 2011)

Em suma, o OM teve (pelo menos em seus primeiros anos) uma intenção voltada para processos focados na continuidade dos sistemas derivados dos estudos eruditos de intervalos, acordes, harmonia que estavam na base das preocupações do serialismo integral. O OM sempre foi um dos software amplamente utilizados para tal fim, e isto reflete diretamente em sua interface e cultura de uso.

O OM é um *framework* que tende a uma programação orientada pela reflexão em tempo diferido, isto é, estimula a composição por escolha entre diversos resultados permutados e decupados em um tempo de escuta.

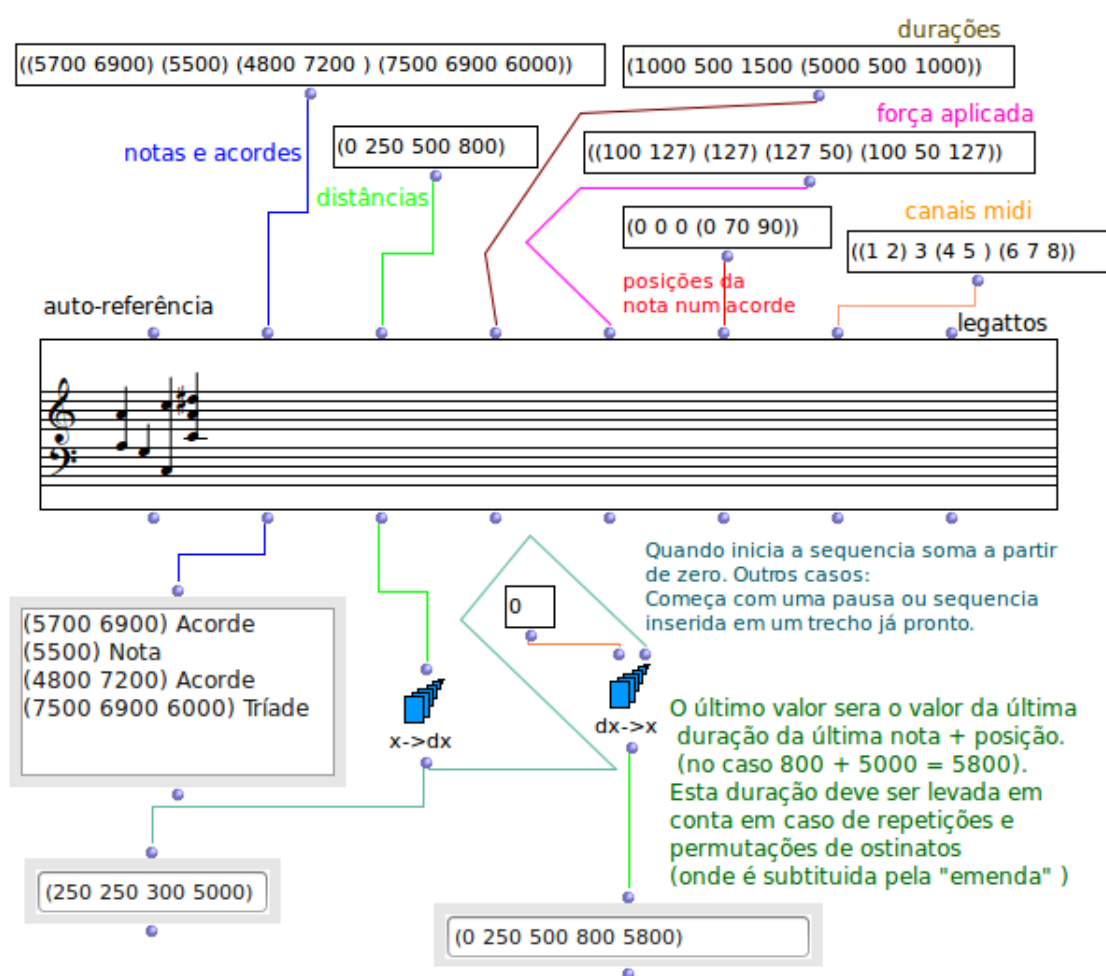
Organiza materiais orientado basicamente pela escrita e fortemente pensado dentro do esquema de intervalos melódicos harmônicos derivados da notação moderna para música orquestral, utilizando sequenciadores bastante similares ao pentagrama de pauta sem (“*chord-seq*”) ou com figuras de compasso (com o sequeciador “*voice*”).

3.1.1.1 chord-seq

A linguagem OM tem objetos avançados para a manipulação e sequenciamento de estruturas em representação partitural. Desde objetos gráficos para representação da nota em pauta, acordes isolados ou mesmo estruturas em árvore para construção de ritmos compostos (“*mktree*”) e polifonia (“*voice*”). Estes objetos podem ser organizados em macro estruturas de sequenciamento, como as “*maquettes*” e “*sheets*” (BRESSION; AGON, 2008).

Importante destacar aqui o funcionamento do objeto **"chord-seq"** para o detalhamento de como opera a lógica dos encadeamentos de acordes, que por trás dos objetos gráficos são entendidos como listas Lisp² aninhadas. Utilizam uma escala de "midicents" onde entre cada semitom da escala cromática é possível também uma divisão de 100 microtons. Por exemplo: ao invés de representar um Dó com o número 12, 24, 48, etc. será representado por 1200, 2400, 4800, etc. Na figura abaixo detalhamos o funcionamento das entradas e saídas de um objeto **"chord-seq"**.

Figura 1 – Objeto Chord-Seq do OM



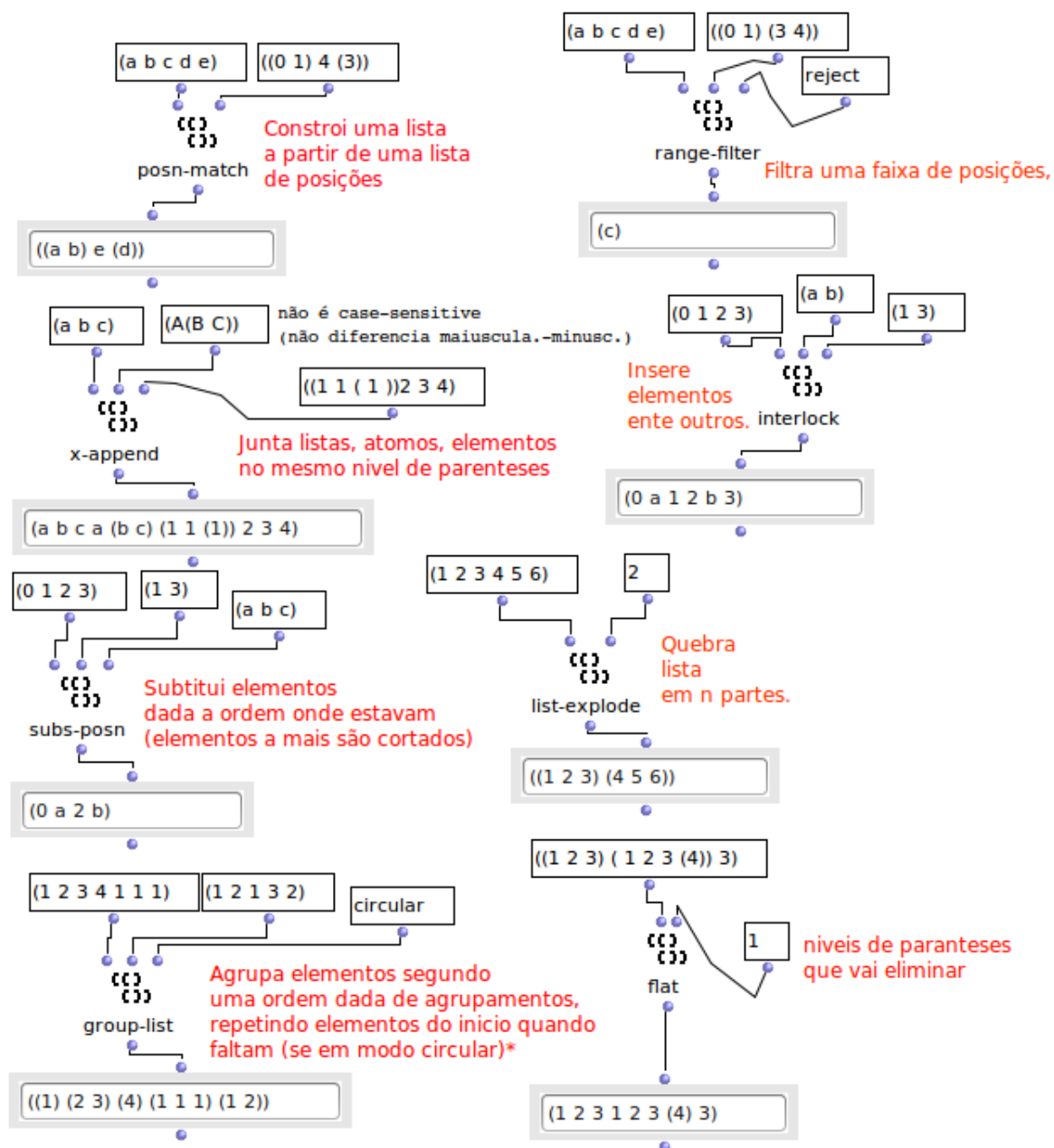
Fonte: autor

² Para detalhes sobre a linguagem de programação LISP, c.f. (GRAHAM, 1995)

3.1.1.2 Listas de dados

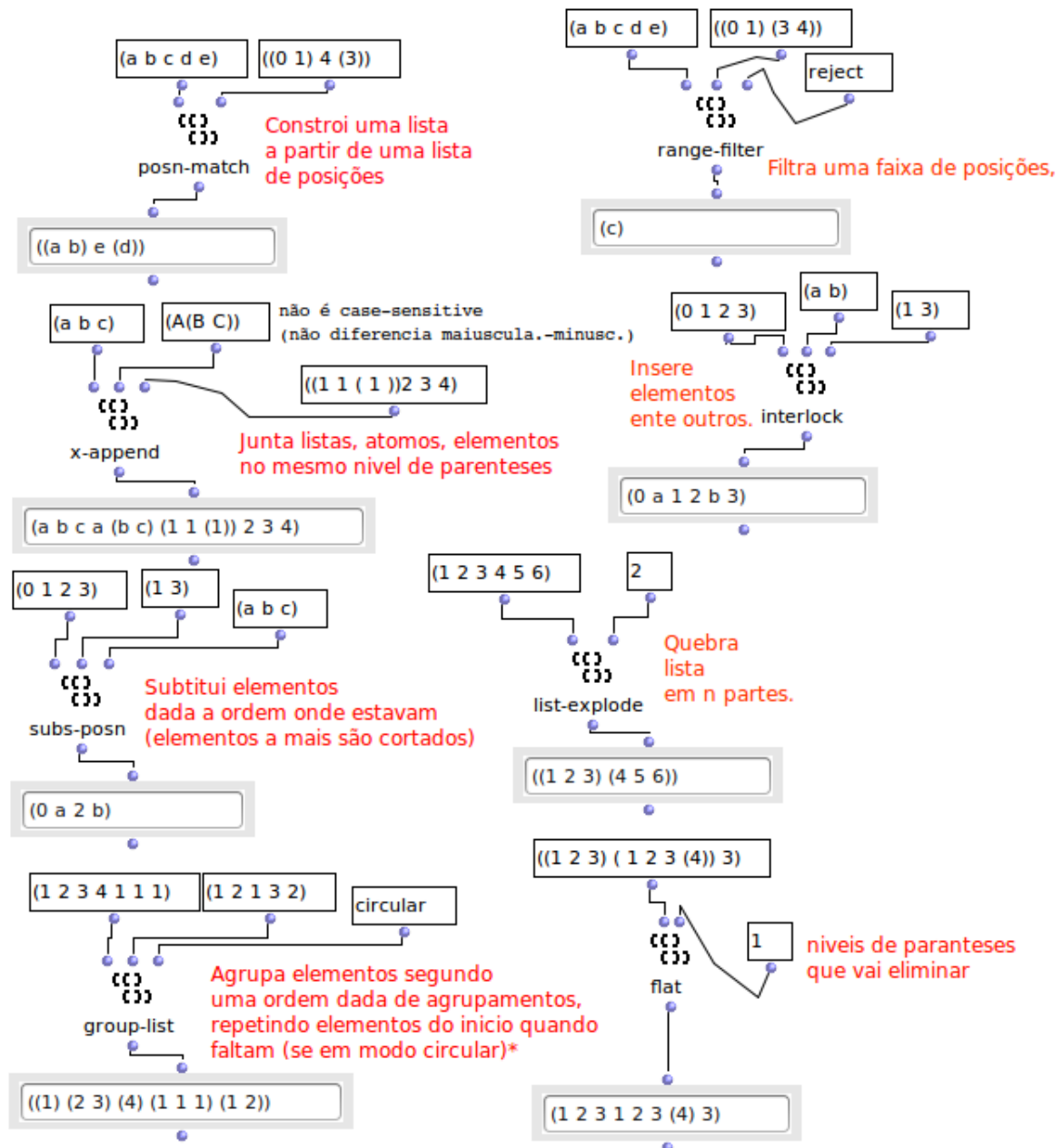
O OM implementa uma série de operações sobre as listas de dados estruturadas. Abaixo selecionamos algumas funções básicas que auxiliam na manipulação dos dados dentro do OM.

Figura 2 – Operações com listas no OMs



Fonte: autor

Figura 3 – Operações com listas no OMs

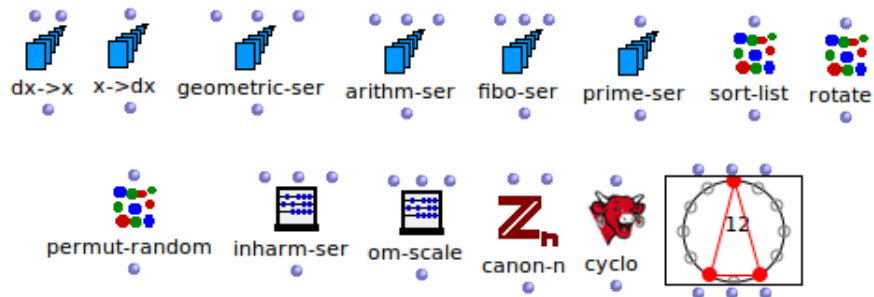


Fonte: autor

3.1.1.3 Operações combinatórias e operações sobre grupos de classes de altura

O OM possui também algumas classes já preparadas especialmente para organizar séries, permutar, ordenar, embaralhar. Há inclusive uma classe chamada *"math"*, que antecipa e organiza uma série de operações básicas sobre grupos de classes de altura, como extração do vetor intervalar, classificação de conjuntos de Allen Forte e uma representação gráfica da geometria das classes de altura, o objeto *"n-cerle"* (ANDREATTA; AGON, 2003). Alguns objetos desta classe foram testados nas demonstrações do ??.

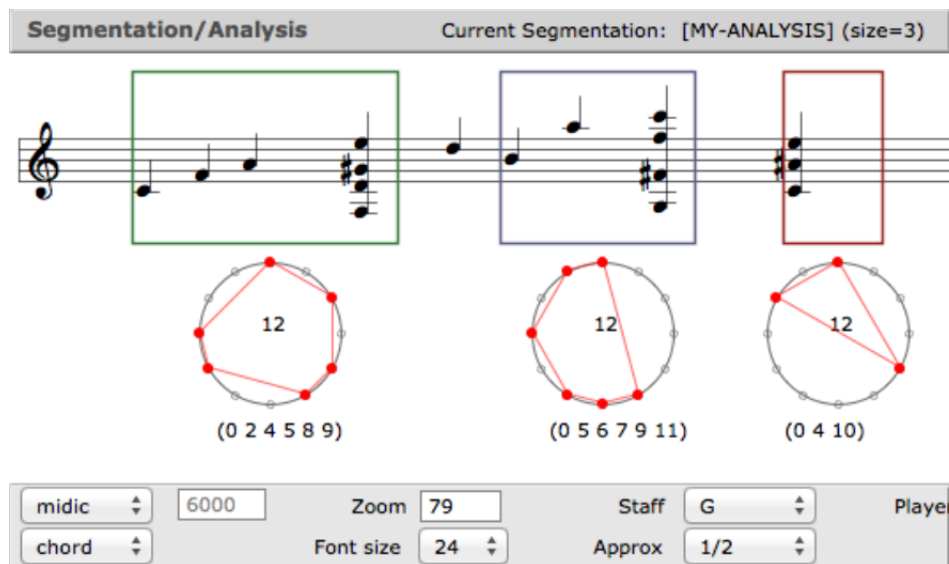
Figura 4 – Alguns objetos para operações combinatórias e operações em grupos de classes de altura no OMs



Fonte: autor

O OM também incorpora dentro de seus sequenciadores de nota um procedimento para segmentar articulações com fins de análise, preparando-os para uma classificação de grupos de alturas com a estrutura de dados do objeto "*n-cercle*" (BRESSON; PÉREZ-SANCHO, 2012).

Figura 5 – Segmentação em Chord-Seq de OpenMusic



Fonte: (BRESSON; PÉREZ-SANCHO, 2012)

3.1.2 PD

Desde o início de seu desenvolvimento o PD surge como um software preocupado em resolver problemas de processamento digital de sinal de maneira estruturada - paralelo ao momento que sua versão comercial *Max*³ desenvolvia sua versão "*MSP*" para processamento de áudio digital, baseada neste primeiro estágio do PD.

Um novo sistema de software, chamado *Puredata*, está em seus primeiros estágios de desenvolvimento. Seu design almeja remediar algumas deficiências do programa Max e preservar algumas de suas vantagens. A mais importante fraqueza do Max é a dificuldade de manter estruturas de dados compostas de um tipo que possa ser acessado quando analisando e resintetizando sons ou quando gravando e modificando sequências de diferentes tipos. Também, tem sido difícil integrar sinais que não sejam de áudio (vídeo por exemplo, e também espectro sonoro) dentro do rígido sistema de "objetos til" (~) do Max. (PUCKETTE et al., 1996)

Interessante perceber que já no final dos anos 90 o PD, tinha como alvo a vindoura possibilidade de uma música performática feita com os computadores domésticos que começavam a se popularizar, preocupando-se com aspectos de melhor performance de processamento de áudio, uso de vídeo sincronizado e representação em tempo real de dados de processamento sonoro, apontando para a possibilidade de outros tipos de representação da música. Seu tutorial nativo é todo focado na didática do processamento digital de sinal. (PUCKETTE, 2007)

A música partiturada em *scores* tradicionais nunca foi uma grande preocupação no PD. Podemos encontrar nativamente uma proposta de estrutura de dados mais próxima dos *scores* eletroacústicos, como o de "Artikulation" de Ligeti ou "Studie II" de Stockhausen.⁴

Apesar disso, vale lembrar também que, assim como o Max, o PD tem uma série de objetos preparados para lidar com o protocolo MIDI. Existem também várias bibliotecas de terceiros (distribuídas no pacote "Extended") que foram criadas pensando em manipulação de dados estruturados como semitons da escala de temperamento igual. Vejamos a seguir.

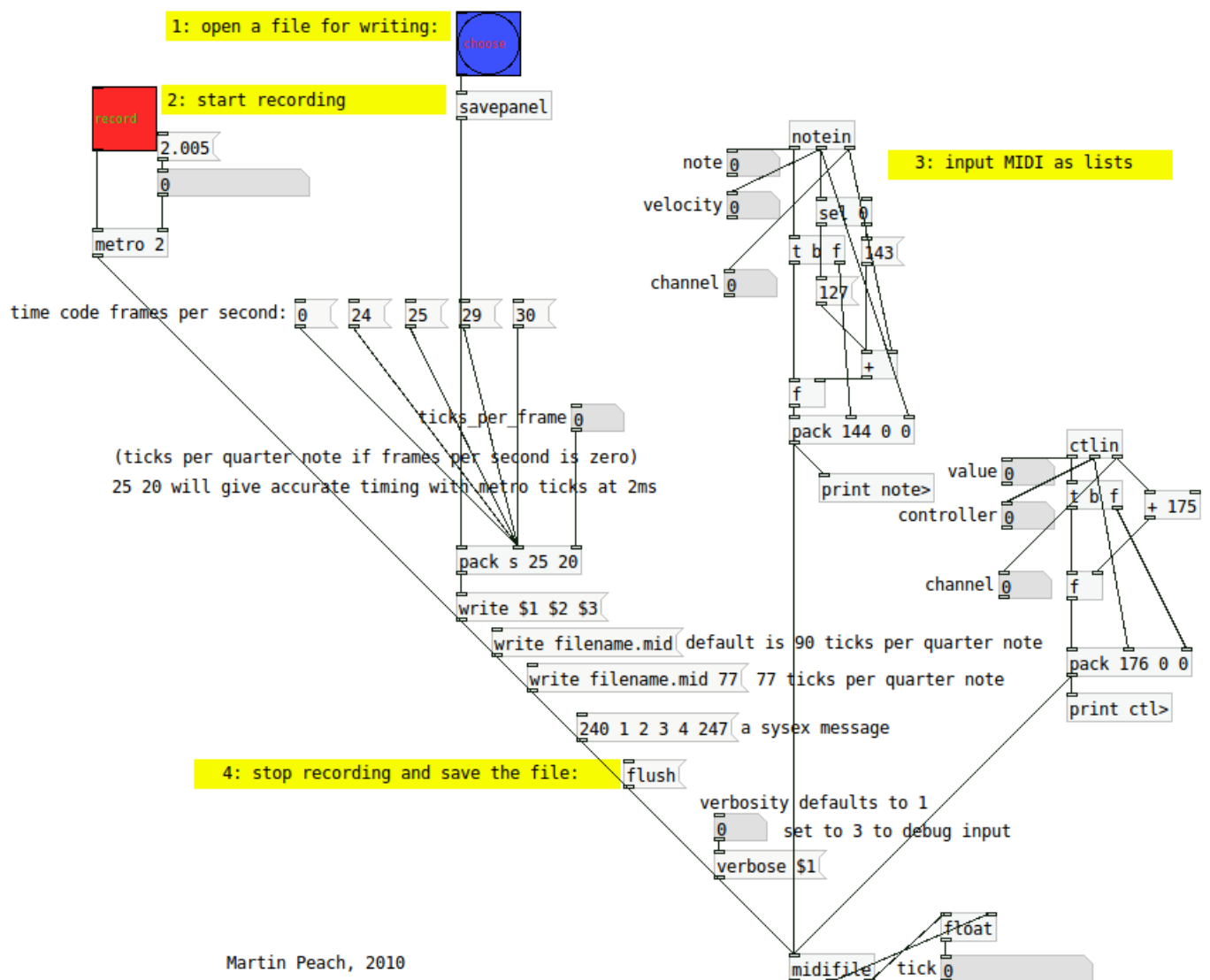
3.1.2.1 Biblioteca [RTC-lib]

A biblioteca "*Real Time Composition*" (RTC-lib) de Karheinz Essl, implementa algumas operações inspiradas em procedimentos seriais, algumas funções inclusive com nomes similares as do OM.

³ <<http://cycling74.com/>> Acesso em 10 de julho de 2014.

⁴ c.f. "Solitude" de Hans C. Steiner: <<http://vimeo.com/16175509>> Acesso em 10 de julho de 2014.

Figura 6 – Manipulação de arquivos MIDI no PD.



Fonte: autor

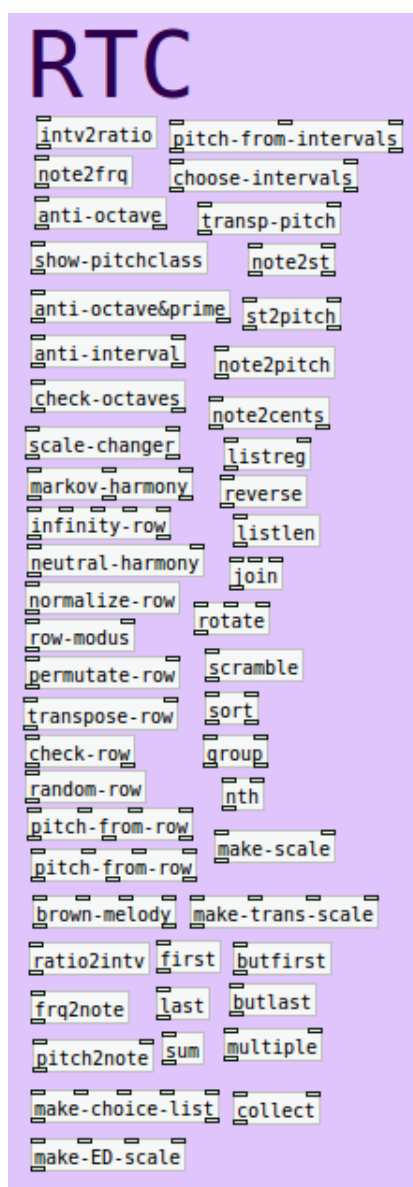
3.1.2.2 Biblioteca de manipulação de listas [list-abs]

Biblioteca que implementa diversas funções para busca, substituição, operações matemáticas, conversões de tipos, permutações e composições de listas de dados em forma de mensagens PD.

3.1.2.3 objeto [probalizer]

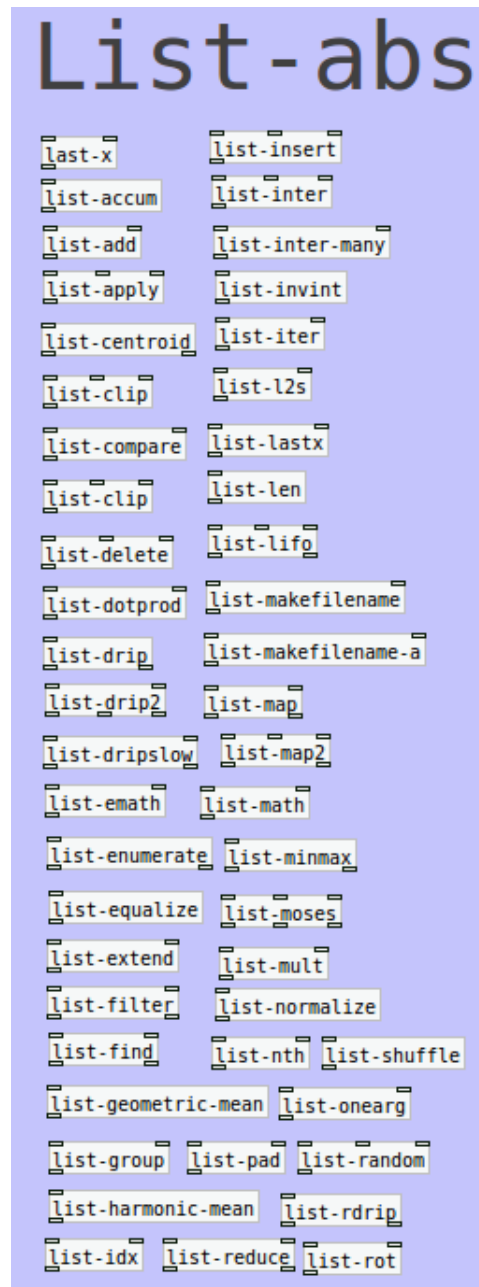
Interessante implementação gráfica de histogramas que podem ser alimentados em tempo real, gerando números dentro da mesma probabilidade, encontrado na biblioteca "Unauthorized".

Figura 7 – Objetos da biblioteca "Real Time Composition" para PD.



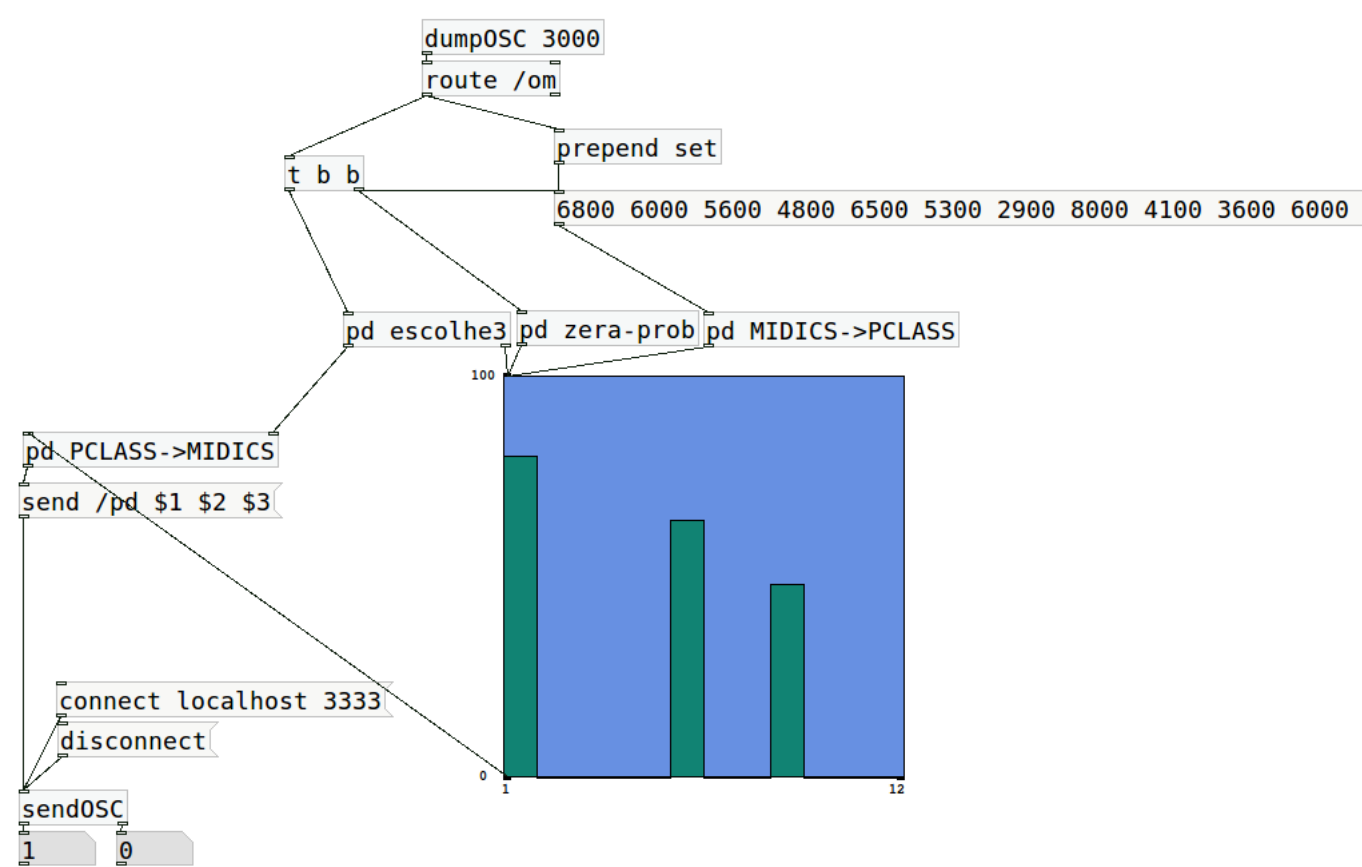
Fonte: autor

Figura 8 – Objetos da biblioteca "list-abs" para PD.



Fonte: autor

Figura 9 – Objeto [probalizer].



Fonte: autor

4 Arquivos e Scripts para Segmentação de Dados Musicais

4.1 MIDI

Por muito tempo o formato MIDI ficou estigmatizado por ser associado aos timbres genéricos da indústria de sintetizadores populares dos anos 80 e 90 e pelas primeiras placas de som e softwares sequenciadores de eventos ou partituras dos computadores pessoais. Na verdade o formato não carrega parâmetros de timbres em seus metadados. Arquivo MIDI carrega valores básicos de expressão sobre a força que a nota deve ser tocada e as alturas cromáticas que devem ser moduladas, permitindo que esta seja posteriormente associada a qualquer timbre.

A mensagem MIDI básica carrega informação sobre:

1. O canal onde vai atuar, permitindo mixar diversos instrumentos em polifonia.
2. O programa que indica o timbre.
3. *NoteOn/NoteOff* - Nota soando e nota sem soar. O MIDI manda duas informações básicas sobre o envelope da nota. Uma primeira nota com a força inicial e uma segunda com a mesma nota e força zero, para silenciá-la.
4. *Velocity* ou expressão: força com qual a nota é tocada.
5. Canal de controle para uso de escala de 127 passos que tem uso dependente da implementação da aplicação. Por exemplo, parâmetros de equalização do timbre. Na prática, o canal de controle é geralmente usado para receber dados de potenciômetros ou sensores analógicos e assinalar a qualquer tipo de parâmetro.
6. *Pitch Bend* - Parâmetro para atuar diretamente na afinação de uma nota em tempo real, de modo similar ao gesto de *bend* de instrumentos de cordas. A especificação MIDI permite que este controle tenha uma granulação de 16.384 pontos e geralmente é usada para um *slide* que cobre duas oitavas.
7. Mensagens exclusivas de sistema (*SysEx*) - geralmente usadas por aplicações para ações independentes do gesto musical. Por exemplo informar ao sistema onde onde buscar arquivos temporários de uma sessão.

É importante ter em mente que o protocolo MIDI, por ser há mais de 35 anos um padrão ainda em uso, gerou um legado relevante de arquivos baseados em repertório clássico

para a reconstituição de corpus de peças partituras. Porém não são descritores capazes de garantir a boa formatação de seus dados como figuras de compasso de uma pauta tradicional, já que os arquivos MIDI não carregam informações sobre as figuras, apenas sobre suas durações, alturas e expressão. Quando importados para programas de notação ou convertidos para formatos destes, os arquivos MIDI irão passar por uma segmentação arbitrária e determinada pelo algoritmo "*parser*"¹ que vai converter determinada duração em determinada métrica quantizada, normalmente diferente das articulações de quais as músicas foram digitalizadas.

Veremos a seguir outros arquivos mais específicos para este fim e quando necessário será feita a conversão entre os tipos.

¹ Método computacional para conversão entre formatos ou tipos de dados diferentes.

4.2 Lilypond

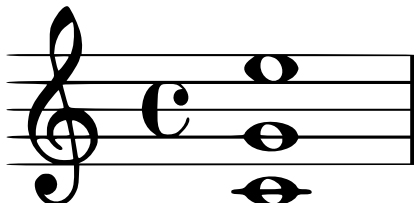
O objetivo principal no Lilypond é a formatação de uma notação partitural avançada e otimizada para impressão em papel. Permite também a utilização de elementos de notação mais exótica, inclusão de texto, dedilhados, nomenclatura de acordes, sinais de expressão, e customização de elementos a partir de módulos. Facilita a otimização da disposição e dimensão das fontes dos objetos e possui uma linguagem *script* própria, dialeto da sintaxe *scheme*².

Figura 10 – Gerador de um acorde Dó maior (dó4 e4 g5) na clave de sol em Lilypond

```
upper = {
  \clef treble
  \key c \major
  \time 4/4

  <c' e' g'>1
}

\score {
  \new PianoStaff <<
    \new Staff = "teste" \upper
  >>
  \layout { }
  \midi { }
}
```



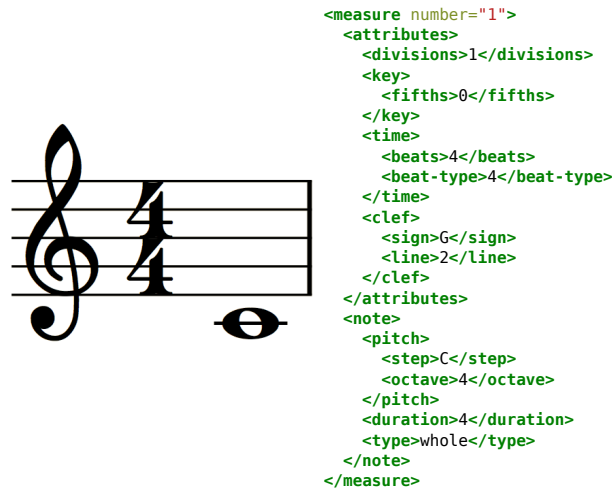
4.3 MusicXML

O uso geral do formato MusicXML é similar ao Lilypond - formatação de partituras. No entanto, enquanto Lilypond é um sistema completo fechado em si próprio, o MusicXML é um formato com a intenção de tornar-se um padrão intercambiável entre diferentes aplicações de partitura³.

² Tutorial oficial de lilypond-scheme: <<http://lilypond.org/doc/v2.16/Documentation/source/Documentation/extending/introduction-to-scheme>> Acesso em 10 de julho de 2014.

³ Lista atualizada de aplicações compatíveis com o formato MusicXML: <<http://www.musicxml.com/software/>> Acesso em 10 de julho de 2014.

Figura 11 – Gerador de uma nota dó4 na clave de sol em MusicXML



4.4 Bibliotecas Python auxiliares

Pela sua natureza de código aberto alto-nível de orientação a objetos, Python (ROSSUM; JR, 1995) é uma linguagem *script* que tem sido amplamente adotada e ampliada por bibliotecas para as mais diversas aplicações científicas e artísticas. (DOWNEY, 2009; KROGER, 2012)

Aliada ao uso de algumas bibliotecas específicas para arquivos de segmentação partitural, Python mostra-se uma ferramenta prática para formatação dinâmica de partituras prontas para impressão ou para auxiliar a análise de dados quantitativos de corpus de partituras.

Falaremos a seguir de duas dessas bibliotecas utilizadas como ferramenta auxiliar nesta pesquisa:

4.4.1 Abjad

É uma biblioteca voltada para a formatação de clichês em notação partitural pronta para impressão em papel, baseada na manipulação de *templates* no formato Lilypond. A biblioteca apresenta alguns templates baseados em peças de Bártok, Ligeti, Ferneyhough e Mozart.⁴

⁴ <<http://abjad.mbrsi.org/examples/index.html>> Acesso em 10 de julho de 2014.

4.4.2 Music 21

É uma biblioteca projetada para trabalhar com manipulação e análise de *corpus* de arquivos partituráveis⁵. Prepara a conversão entre diversos arquivos de dados musicais (MIDIs, humdrum, lilypond, abc)⁶, mas nativamente trabalha com uma estrutura de dados baseada em Music XML.

Music21 tem uma abordagem voltada para uma "musicologia assistida por computador" e já tem incorporada em suas classes algumas ferramentas comuns a esta prática como: numeração de grau funcional de acorde⁷, numeração de classes de altura usando a classificação de Allen Forte⁸ e a implementação dos algoritmos de detecção de tonalidade⁹ elaborado por Krumhansl (1990) e aperfeiçoado por Temperley (2001), descritos nesta pesquisa.¹⁰

⁵ <<http://web.mit.edu/music21/doc/moduleReference/moduleCorpus.html>> Acesso em 10 de julho de 2014.

⁶ <<http://web.mit.edu/music21/doc/moduleReference/moduleConverter.html>> Acesso em 10 de julho de 2014.

⁷ <<http://web.mit.edu/music21/doc/moduleReference/moduleRoman.html>> Acesso em 10 de julho de 2014.

⁸ <<http://web.mit.edu/music21/doc/moduleReference/moduleChord.html?#music21.chord.Chord.forteClassNumber>> Acesso em 10 de julho de 2014.

⁹ <<http://web.mit.edu/music21/doc/moduleReference/moduleAnalysisDiscrete.html>> Acesso em 10 de julho de 2014.

¹⁰ ??

Parte III

Experimentos Composicionais Generativos

5 Gerador de MikroBagatellas

6 CosmoSonoridade de um Rascunho em Aberto

Proponho aqui um experimento mais livre,

Conclusão

A busca por regras gerativas (ROADS, 1979) que pudessem formalizar algoritmos a partir de análises musicais de dados quantitativos capazes de segmentar e isolar parâmetros nos levou a encontrar uma interessante diferença entre duas abordagens analíticas que nos parecem complementares, apesar de contraditórias em certos aspectos.

Por um lado, revisamos uma corrente teórica que enumera estruturas operantes nas expectativas da música tonal. As argumentações derivadas da "Teoria Gerativa da Música Tonal" (GTTM) operam sobre aquele repertório considerado como prática comum nas análises funcionais das cadências, modulações, prolongamentos, tensão e relaxamento. Tais teorias argumentam que esta normatização é derivada de um condicionamento cultural na escuta da música ocidental e procuram justificar suas regras de "boa-formação" ou "preferência" (LERDAHL; JACKENDOFF, 1983; TEMPERLEY, 2001) legitimando as fórmulas a partir de teses linguísticas das estruturas de sintaxe derivadas da fala e escrita (CHOMSKY, 1957) e as pesquisas de campo em cognição musical básica das relações entre intervalos e percepção subjetiva das alturas em seu espaço tonal inferido pelo ouvinte (KRUMHANSL, 1990; LERDAHL, 2001).

Por outro lado nos chamou atenção que em paralelo a fundamentação destas teorias, nas últimas décadas do século XX¹ formalizou-se uma "teoria de grupos das classes de altura" (FORTE, 1973; RAHN, 1980; PERLE, 1990; STRAUS, 2004) baseada numa catalogação de combinações de intervalos formando compostos sonoros singulares e as possíveis estruturações e articulações entre estes. Ali operariam critérios não necessariamente tão amarrados na funcionalidade dos esquemas de tensão e relaxamento das formas "tonalizantes". Mesmo que vertiginosamente, é preciso admitir que cada composição já pode comportar um sistema totalmente idiossincrático. Ao contrário da corrente cognitivista, neste caso uma criatividade autoral do analista assume que as formas estão emergindo ali a princípio porque foram apontadas, e não necessariamente porque foram intuídas pelo compositor. Também não buscam justificar expectativas do ouvinte que guiariam a normatização de uma busca composicional da "boa forma" pré-concebida (BABBITT, 1958).

Dadas estas duas perspectivas pretendemos continuar este trabalho com uma organização dos algoritmos sugeridos por estas em bibliotecas das linguagens de programação apontadas na Parte II, incluindo aí aprofundar e apontar boas práticas nas linguagens almejando uso otimizado destas teorias em processos composicionais. Nos interessa testar os limites práticos destas implementações, deixando um legado tecnicamente apropriável,

¹ A partir da sedimentação de uma tradição pós-tonal iniciada nas primeiras décadas do século XX.

utilizando ferramentas e bibliotecas auxiliares disponíveis em software livre.

Utilizaremos para exemplificar o uso das regras observadas em contextos tonais e pós-tonais alguns aspectos dos algoritmos atuando em um corpus de peças do repertório da suíte Mikrokosmos de Béla Bartók que será revisado no formato *Music XML* para o estudo computacional, buscando respeitar a grafia das partituras originais.

Partiremos de pistas já deixadas por autores que aprofundaram o tema (MARSHALL, 1946; SUCHOFF, 1971; LENDVAI; BUSH, 1971; ANTOKOLETZ, 1984; SUCHOFF, 2004; LESTER, 1989).

Mais do que encontrar alguma nova abordagem analítica sobre estas peças, buscaremos a demonstração de algumas regras gerativas observadas, para em seguida serem utilizados de maneira mais livre em procedimentos composicionais que gerem motivos partiturados - e sugestões de encadeamento destes.

Composições produzidas durante esta pesquisa estão disponíveis para *download*². Também estão disponíveis os códigos que estão sendo trabalhados³. A próxima etapa será modularizar e comentar os códigos, detalhando os processos da maneira mais didática possível.

Vale lembrar que estamos convencidos de que uma abordagem analítica que desconside parâmetros determinantes sobre a sonoridade (GUIGUE, 2012) dos aglomerados de alturas, como parte intencional e determinante na composição estará incompleta. Também acreditamos que é limitador ignorar a tipo-morfologia (SCHAEFFER, 1966) e espectro-morfologia (SMALLEY, 1997) dos timbres atuantes numa situação real de gravação, performance partiturada ou improvisado sobre as estruturas geradas. Interessa-nos refletir sobre estes aspectos em alguma continuidade deste trabalho.

² <<http://vimeo.com/16175509>> Acesso em 10 de julho de 2014.

³ <<https://github.com/glerm/luteriaOM>>, <https://github.com/glerm/Derivas_OpenMusic> e <<https://github.com/glerm/AutomatosGeradores>> Acesso em 10 de julho de 2014.

Referências

- AGON, C. *The OM Composer's book 1*. Musique/sciences. Delatour France, 2006. ISBN 9782752100283. Disponível em: <<http://amazon.com/o/ASIN/2752100280/>>. Citado na página 27.
- ANDREATTA, M.; AGON, C. Implementing algebraic methods in openmusic. In: *Proceedings of the International Computer Music Conference, Singapore*. [S.l.: s.n.], 2003. Citado na página 31.
- ANTOKOLETZ, E. *The music of Béla Bartók: a study of tonality and progression in twentieth-century music*. [S.l.]: Univ of California Press, 1984. Citado na página 52.
- BABBITT, M. Who cares if you listen? *High Fidelity*, v. 8, n. 2, p. 38–40, 1958. Citado na página 51.
- BACHMANN, T.; BACHMANN, P. J. An analysis of bela bartok's music through fibonaccian numbers and the golden mean. *Musical Quarterly*, JSTOR, p. 72–82, 1979. Citado na página 22.
- BRESSON, J.; AGON, C. Scores, programs, and time representation: The sheet object in openmusic. *Computer Music Journal*, MIT Press, v. 32, n. 4, p. 31–47, 2008. Citado na página 28.
- BRESSON, J.; AGON, C.; ASSAYAG, G. Openmusic: visual programming environment for music composition, analysis and research. In: ACM. *Proceedings of the 19th ACM international conference on Multimedia*. [S.l.], 2011. p. 743–746. Citado na página 28.
- BRESSON, J.; PÉREZ-SANCHO, C. New framework for score segmentation and analysis in openmusic. *Proceedings Sound and Music Computing Conference*, Copenhagen, Denmark, 2012. Citado na página 32.
- CHOMSKY, N. Syntactic structures. Mouton, 1957. Citado 2 vezes nas páginas 14 e 51.
- COHN, R. Bartok's octatonic strategies: A motivic approach. *Journal of the American musicological society*, JSTOR, p. 262–300, 1991. Citado na página 21.
- DOWNEY, A. *Python for software design: how to think like a computer scientist*. [S.l.]: Cambridge University Press, 2009. Citado na página 42.
- FILHO, W. N. *Os voos do Passarinho de Pano e análise dos processos composicionais na suíte Prole do Bebê nº 2 de Villa-Lobos*. Tese (Doutorado) — Universidade de São Paulo, 2012. Citado na página 19.
- FORTE, A. *The structure of atonal music*. [S.l.]: Yale University Press, 1973. Citado 3 vezes nas páginas 5, 14 e 51.
- GRAHAM, P. *ANSI common lisp*. [S.l.]: Prentice Hall Press, 1995. Citado na página 29.
- GUIGUE, D. *Estética da Sonoridade*. São Paulo: Editora Perspectiva, 2012. Citado na página 52.

- IAZZETTA, F. *Música e mediação tecnológica*. São Paulo: Perspectiva-Fapesp, 2009. Citado na página 13.
- JOHNSON, T. Found mathematical objects. *samedi*, v. 13, p. 17, 2001. Citado na página 19.
- KELLER, R. C. *Rotational Strategy in Selected Works by Bela Bartok*. Tese (Doutorado) — Florida State University, 2011. Citado na página 21.
- KROGER, P. *Music for Geeks and Nerds*. CreateSpace Independent Publishing Platform, 2012. ISBN 9781478345381. Disponível em: <<http://musicforgeeksandnerds.com>>. Citado na página 42.
- KRUMHANS, C. L. *Cognitive foundations of musical pitch*. [S.l.]: Oxford University Press New York, 1990. Citado 3 vezes nas páginas 14, 43 e 51.
- LENDVAI, E. *Duality and synthesis in the music of Béla Bartók*. [S.l.: s.n.], 1962. Citado na página 21.
- LENDVAI, E.; BUSH, A. *Béla Bartók: an analysis of his music*. [S.l.]: Kahn & Averill London, 1971. Citado 2 vezes nas páginas 22 e 52.
- LERDAHL, F. Tonal pitch space. *Music Perception: An Interdisciplinary Journal*, University of California Press, v. 5, n. 3, p. pp. 315–349, 1988. ISSN 07307829. Disponível em: <<http://www.jstor.org/stable/40285402>>. Citado na página 9.
- LERDAHL, F. *Tonal pitch space*. [S.l.]: Oxford University Press, 2001. Citado na página 51.
- LERDAHL, F. Genesis and architecture of the gttm project. JSTOR, 2009. Citado na página 5.
- LERDAHL, F.; JACKENDOFF, R. S. *A generative theory of tonal music*. [S.l.]: MIT press, 1983. Citado 4 vezes nas páginas 5, 9, 14 e 51.
- LESTER, J. Analytic approaches to twentieth-century music. WW Norton & Company, 1989. Citado 2 vezes nas páginas 14 e 52.
- MARSHALL, W. E. *An Analysis Of The Mikrokosmos Of Béla Bartók*. Tese (Doutorado), 1946. Citado na página 52.
- MORGAN, R. P.; ANTOKOLETZ, E. *Twentieth-century music: a history of musical style in modern Europe and America*. [S.l.]: Norton New York, 1991. Citado na página 21.
- PERLE, G. Pitch-class set analysis: An evaluation. *Journal of Musicology*, JSTOR, p. 151–172, 1990. Citado na página 51.
- PUCKETTE, M. The om composer's book 1. In: _____. Paris, France: Delatour, 2006. cap. Computing while composing, p. x–xiv. Citado na página 27.
- PUCKETTE, M. *The Theory and Technique of Electronic Music*. World Scientific Publishing Company, 2007. ISBN 9789812700773. Disponível em: <<http://amazon.com/o/ASIN/9812700773/>>. Citado na página 33.

- PUCKETTE, M. et al. Pure data: another integrated computer music environment. *Proceedings of the Second Intercollege Computer Music Concerts*, Citeseer, p. 37–41, 1996. Citado na página 33.
- RAHN, J. *Basic atonal theory*. [S.l.]: Longman New York, 1980. Citado na página 51.
- ROADS, C. Grammars as representations for music. *Computer Music Journal*, JSTOR, p. 48–55, 1979. Citado na página 51.
- ROADS, C. *The computer music tutorial*. [S.l.]: MIT press, 1996. Citado na página 27.
- ROADS, C. *Microsound*. [S.l.]: MIT press, 2004. Citado na página 27.
- ROIG-FRANCOLÍ, M. A. *Understanding post-tonal music*. [S.l.]: McGraw-Hill, 2008. Citado na página 19.
- ROSSUM, G. V.; JR, F. L. D. *Python reference manual*. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995. Citado na página 42.
- SCHAEFFER, P. *Traite des objets musicaux. Essai interdisciplines*. Paris: Ed. du Seuil, 1966. Citado na página 52.
- SETHARES, W. *Tuning, Timbre, Spectrum, Scale*. [S.l.]: Springer, 2005. ISBN 9781852337971. Citado na página 27.
- SMALLEY, D. Spectromorphology: explaining sound-shapes. *Organised sound*, Cambridge Univ Press, v. 2, n. 02, p. 107–126, 1997. Citado na página 52.
- STRAUS, J. N. *Introduction to Post-Tonal Theory (3rd Edition)*. [S.l.]: Pearson, 2004. Citado 3 vezes nas páginas 5, 14 e 51.
- SUCHOFF, B. *Guide to Bartók's Mikrokosmos*. [S.l.]: Boosey and Hawkes, 1971. Citado na página 52.
- SUCHOFF, B. *Bartók's Mikrokosmos: Genesis, Pedagogy, and Style*. [S.l.]: Scarecrow Press, 2004. Citado na página 52.
- TEMPERLEY, D. *The cognition of basic musical structures*. [S.l.]: MIT press, 2001. Citado 5 vezes nas páginas 5, 9, 14, 43 e 51.