Rewriting & Music

11th International School on Rewriting
Paris, MINES ParisTech, 1-6 July 2019

florent.jacquemard@inria.fr

Plan

<u>part 0</u>. (today) Examples in Musical Creation at different Representation Levels — acoustic/physical domain & notated/symbolic domain

<u>part I</u>. (today) Sequential Music Representations — notated/symbolic domain
Melodic Similarity, Computational Musicology
Weighted String Rewriting Systems & Edit Distances

<u>part 2</u>. (tomorrow) Tree-structured Music Representations — notated/symbolic domain
Music Notation Processing, Transcription
Term Rewriting Systems & Weighted Tree Automata

(click on a part to jump to its first slide)

Part II

Hierarchical Representations of Music Notation
Music Transcription & Score Processing

Term Rewriting
Weighted Tree Automata

with
Vertigo team, CNAM Paris, Philippe Rigaux
Nagoya University, Masahiko Sakai
Ircam, Paris, Jean Bresson

Plan

1. (digital) music scores

2. Tree-structured representations of music notation
   Rhythm Trees : a hierarchical representation of time

3. Rewriting theory of rhythm notation

4. Rhythm Tree languages & Enumeration

5. Application to Automated Music Transcription

Digital Music Scores

Besides Audio Data:

Processing & Information Retrieval for **Music Notation**?

composition class of Henri Büsser, National Music Conservatory of Paris, 1945

music notation: an essential vector of transmission in Western musical practice
a means of preserving cultural heritage

Music Notation for Music Practitioners

Music notation = graphical format for music data
since ~1000 (Guido d'Arezzo)

(digital) music scores, a tool for

- composers
  authoring, exchange
- performers
  performance : real-time reading or memoization
- editors
  online digital score libraries *e.g.* nkoda.com
- teachers & students
  transmission
- librarians
  heritage : *e.g.* Gallica
- scholars (historians, musicologists...)
  research, analysis

Philippe Manoury - Tensio

for string quartet and electronic (2010)

http://www.philippemanoury.com

virtual quartet (electronics)

this 3 staves are written in traditional music notation (instead of a DSL for sound processing), in order to express synchronisation with the parts of the string quartet consistently.

real string quartet

Digital Music Scores for musicology

digital music scores often contain PDF files (online stores *etc*)

XML score formats emerged in 2000's (MusicXML, MEI...)

they enable search and retrieval by content
for scholars, corpus analysis by digital musicology
(statistics, classification, similarity evaluation)
on individual scores or databases *cf.* Music 21 (MIT)

digitalisation (from paper scores) :
Optical Music Recognition (OMR) or automated music transcription

Digital Scores for Musical Performance, Teaching, Mediation

iPad displays (stands) for music ensembles
annotation, synchronisation, archiving...

Brussells' Philarmonics
using NeoScores app

players
- MIDI
- multi-modal
  (alignment score/audio)

The Sheet Music Interface
for multimodal music presentation and navigation

SmartMusic “tutor with infinite patience”

score following (realtime alignment)
for instruments’ teaching (with feedback)
or automatic accompaniment

Digital Music Scores, accessibility

digital (XML) scores can be modified by musicians (performers)

- page skip, arrangements (*e.g.* ossia),
- notation (fingering, synchronization instructions…),
- adaptations for accessibility (magnify fonts, coloured notes, Braille),
- for gamers, visual artists…

copyright-free scores

Score authorship belongs to composers <u>and</u> editors → limitations for copying and sharing

crowdsourcing project OpenScore involving

IMSLP Petrucci Music Library database of free scores (mostly PDF scans)

musescore free and open source (GPL) music edition software

Tree Representations in Music Information Retrieval and Music Notation Processing

Tree representations in Jazz Harmonic Analysis

Daniel Harasim, Martin Rohrmeier, Timothy J. O'Donnel
A Generalized Parsing Framework For Generative Models Of Harmonic Syntax
ISMIR 2018, Journal of Mathematics and Music 5(1) 2011

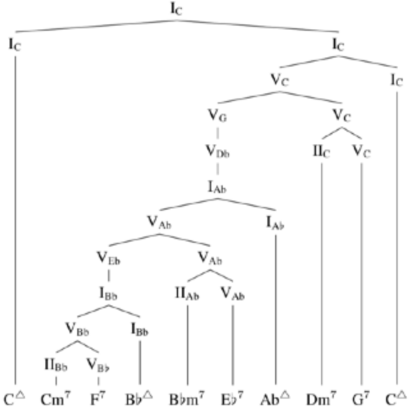


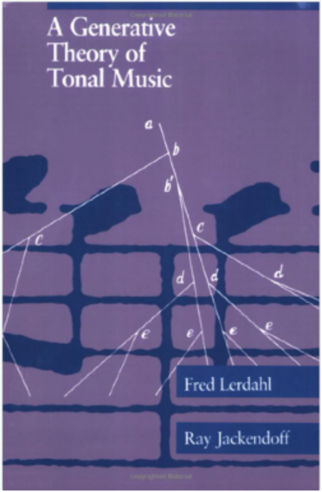
Figure 1. Hierarchical analysis of the A-part of the Jazz-standard *Afternoon in Paris*.

Tree representations of Shenkerian Analysis

Fred Lerdahl, Ray S. Jackendoff
GTTM
MIT press, 1983

tree representations of Shenkerian analyses

Generative Theory of Hip-Hop
Jonah Katz (MIT)



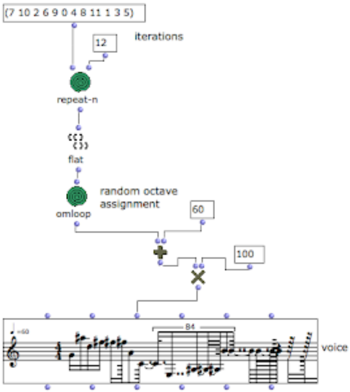
Open Music Rhythm Trees

intermediate representation of rhythms in
in OpenMusic, a LISP programming graphical environment for assisted composition

Michael Laurson
Patchwork: A Visual Programming Language
Helsinki: Sibelius Academy, 1996

Carlos Agon, Karim Haddad, Gérard Assayag
Representation and Rendering of Rhythm Structures
JIM, 2002

Rizo
Symbolic music comparison with tree data structures
PhD thesis U. Alicante, 2010



algorithmiccomposer.blogspot.com

Rhythm Notation & Meter

Beat Hierarchies

Beat Making hardware

16 beats

= 2 * 8 beats

= 4 * 4 beats (quadruple meter)

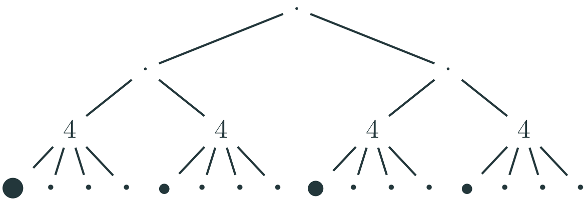
meter

```
x   x   x   x
x x x x x x x x
xxxxxxxxxxxxxxxx
```

strong beat (accent)   weak beat

Example beatmaking

Tree representation of a 4×4 meter for the previous beatmaking hardware – the 16 beats are in the leaves.



size of dot $=$ metrical strength

in musical notation, 16 beats in 4 measures separated by barlines (time signature $=$ 4/4 $=$ 4 beats, also denoted by 'C'):

barline

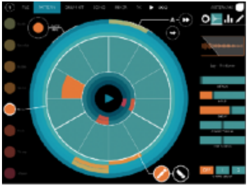
Beats Hierarchies (2)

Beat Making software
Patterning Drum Machine

4 * 4 beats

meter

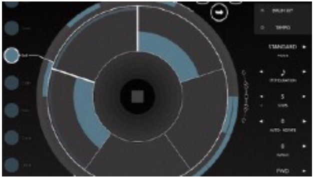
```
x   x   x   x
x x x x x x x x
xxxxxxxxxxxxxxxx
```



3 * 4 beats

meter

```
x   x   x
x x x x x x
xxxxxxxxxxxx
```



5 beats

Meter

The meter is a hierarchical organization of time,
with regularly recurring patterns of strong and weak beats (accents).

Think of

- steps in dances
- greek/latin antic poetry (inspired the notion of meter in Western music notation)

Beats are time positions.

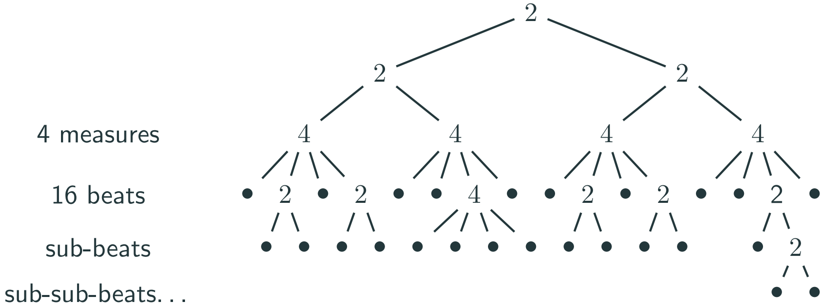
Strong beats may or may not correspond to events.
Strong beat without an event = surprise
(syncopation, because it is against listener's expectation).

Simple Quadruple Meter

A time signature defines a meter.

example *beatmakers*: 16 beats organized in 4 measures
quadruple meter, time signature 4/4:

- each measure contains 4 beats
- each beat can be subdivided by 2 (simple meter)
- and nested subdivisions by 2 *etc*.



Simple Meters (2)

Similarly for simple triple meters (*e.g.* time signatures 3/2, 3/4, 3/8)

- each measure contains 3 beats
- each beat can be subdivided by 2
- and nested subdivisions by 2 *etc.*

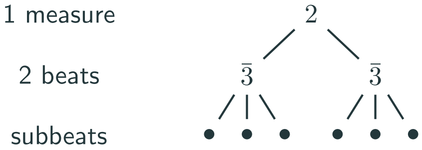
There are also simple duple meters (*e.g.* time signature 2/2, 2/4, 2/8), quintuple meters, septuple meters, *etc.*

Compound Meters

Time signature $6/8 =$ compound duple meter

each measure contains 2 beats,

each beat can be subdivided by 3 (compound meter).

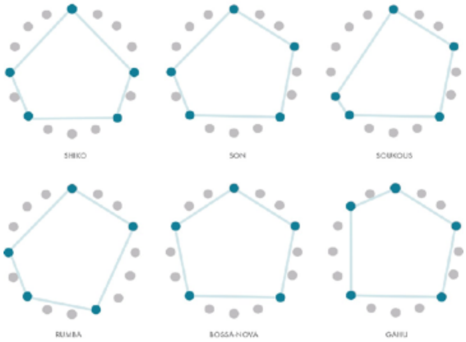


Similarly: compound triple metre (time signature 9/8) compound quadruple metre (time signature 12/8)...

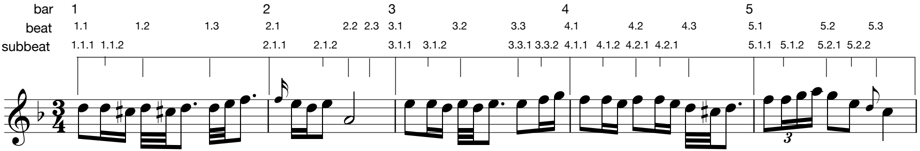
Euclidian Rhythms

The Geometry of Musical Rhythm:
What Makes a "Good" Rhythm Good?
Godfried Toussaint
CRC Press

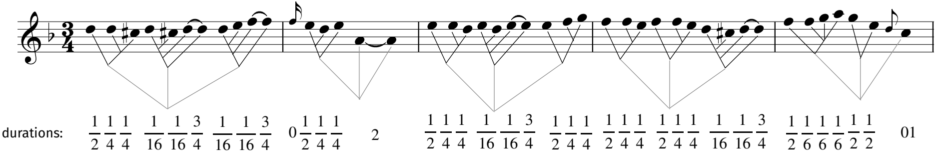
The distance geometry of music
Godfried Toussaint et al.
Computational Geometry 42 (2009) 429–454



The notation gives clues (to player) of the metric structure



Polonaise in D minor from Notebook for Anna Magdalena Bach   BWV Anh II 128

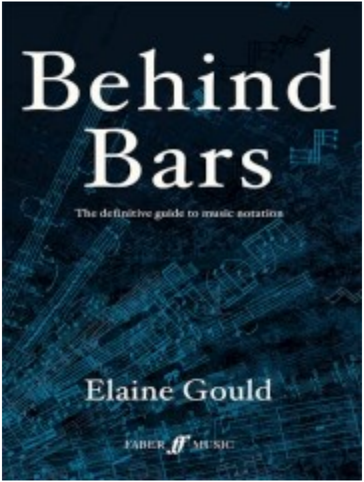


Term Rewriter’s rhythmic notation

with hierarchical encoding of durations: “*the (duration) data is in the structure*”

- the tree leaves contain the events
- the branching define durations, by uniform division of time intervals

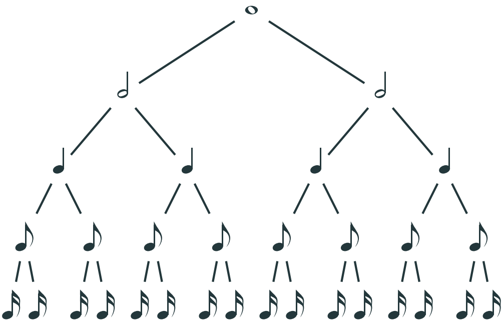
Rhythm Trees



Behind Bars: The Definitive Guide to Music Notation
Elaine Gould
Faber Music

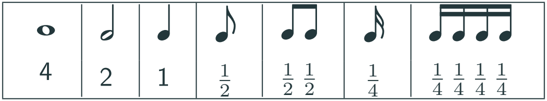
Hierarchical Notation of Time

In Common Western Music Notation, duration are expressed hierarchically, by nested divisions (of measures, beats, *etc.*)

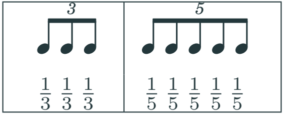


Durations in Common Western Music Notation

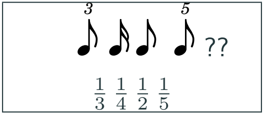
Notation of individual notes (with different note heads and flags) and groups of notes (with beams)



other "irregular" groups, *e.g.* triplet & quintuplet in simple meters:



not every sequence of durations can acceptably be written



Signature Σ:

constant symbols:

$\bullet$: 1 note event

binary symbols:

2 : binary division of time interval, without beam,
$\bar{2}$: binary division with beam

ternary symbols:

3 : ternary division of time interval, without beam,
$\bar{3}$: ternary division with beam.

Rhythm Trees

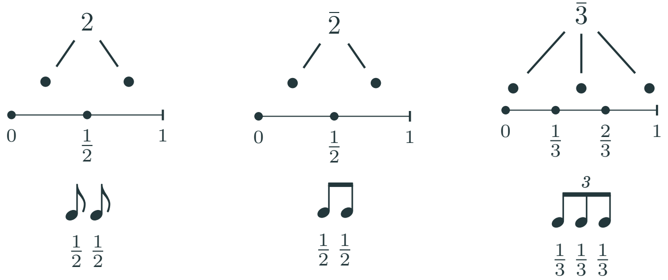
A Rhythm Tree is a ground term in $\mathcal{T}(\Sigma)$.

RT first examples

Beams (ligatures) are horizontal lines connecting notes, substituting the individual flags (with same meaning for durations).

They are used for grouping, in order to

- simplify the reading of notation, and
- highlight the meter.

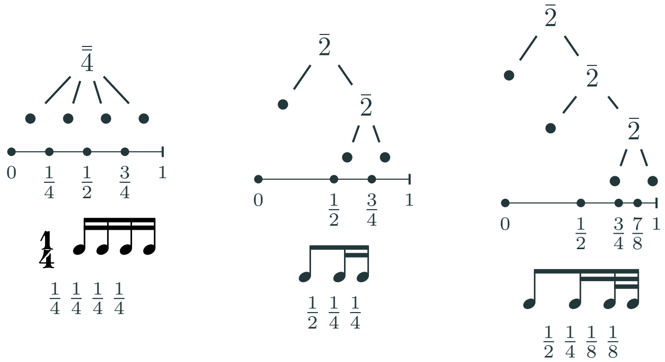


Note the mark *'3'* (= 3:2) for the triplet in simple meters.

Nested RTs

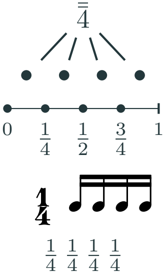
Tree nesting is denoted with beaming

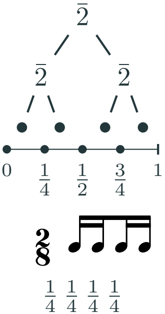
roughly: depth = number of beams/flags



RT nesting

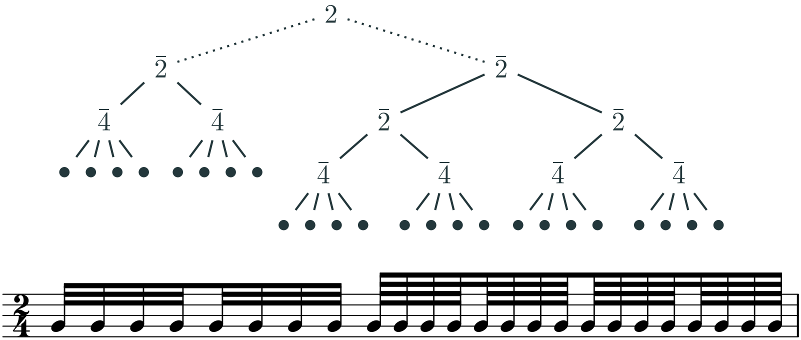
Different nesting/beamings for different meters.





Note the broken secondary beams dividing the grouping,
It indicates metric separation and eases reading.

Broken secondary beams



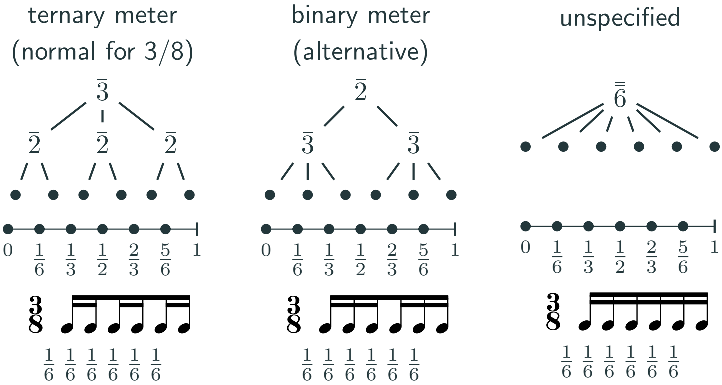
The number of beams separating two subgroups must be proportional to the duration of the groups they separate.

Helen Gould Behinds Bar

Beaming & Meter

Broken beams can also be used to suggest an alternative meter (to the meter defined by time signature).

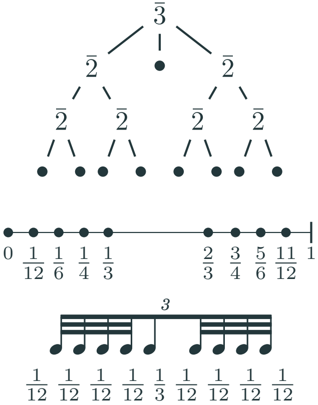
In this example, the time signature is $3/8$ = simple triple meter.

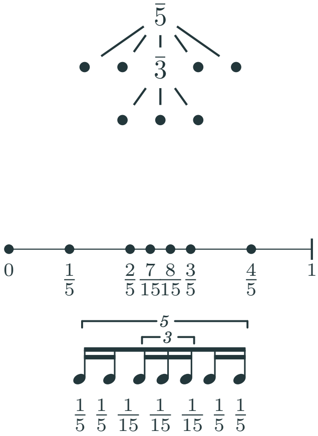


Nested Tuplets

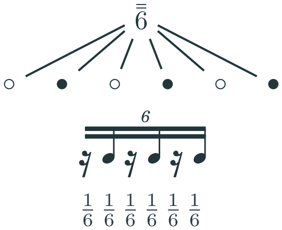
Special tuplet notation, for divisions not corresponding to the meter.

They can be nested.



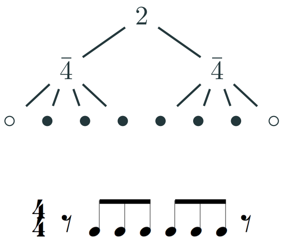


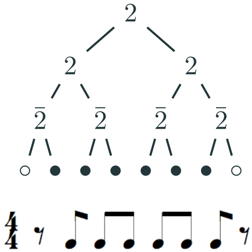
Rests (silence) are events in the Rhythm Tree encoding, they are represented by a constant symbol ∘.

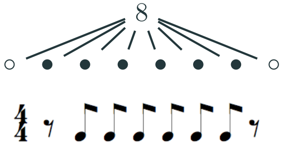


Beaming in simple quadruple meter

Various notations for $\left[\frac{1}{2}\right]\ \frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}\ \left[\frac{1}{2}\right]$ in a 4/4 time signature.



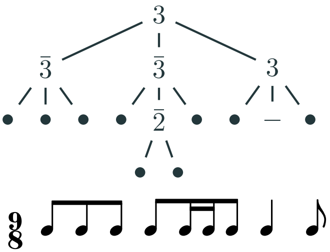




By convention, in the 4/4 time signature there are never beams between the notes of beat 2 and of beat 3, in order to keep visible the middle point of the measure.

Beaming in a compound triple meter

Time signature 9/8: 3 beats per measure, each beat is divided into 3.



Grace notes

A grace note is an out-of-time event, with duration zero.

Signature Σ:

constant symbols:

$\circ$ = 1 rest event

$\bullet$ = 1 note event

$\bullet_1$ = 1 grace note followed by 1 note event

$\bullet_2$ = 2 grace notes followed by 1 note event

...

binary symbols:

2 (binary division without beam),

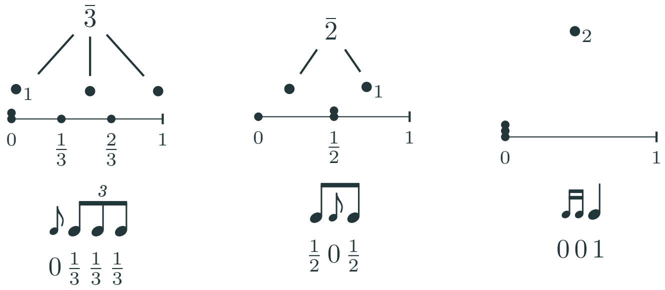
$\bar{2}$ (binary division with beam)

...

- fix bounds for a finite signature.

- extended frameworks for infinite signature.

Grace notes (2)

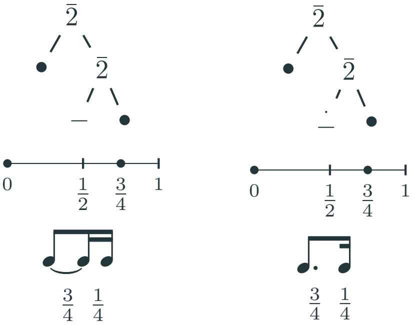


practical interpretation: 'play as you wish'.

Such loose semantics makes it difficulties to handle grace note in transcription. Without care, all events could be transcribed as grace-notes!

Ties and Dots

A leaf labeled by – (tie) augments the duration of the previous leaf.
Hence it represents no new event.



Ties and Dots (2)

RT: durations of positions

For a rhythm tree t,

$$\mathrm{dur}(\mathrm{root}) \;=\; 1 \text{ beat}$$

if p is a leaf, $p \neq \mathrm{root}(t)$, $\mathrm{nextleaf}(p)$ exists and $t(\mathrm{nextleaf}(p)) = -$

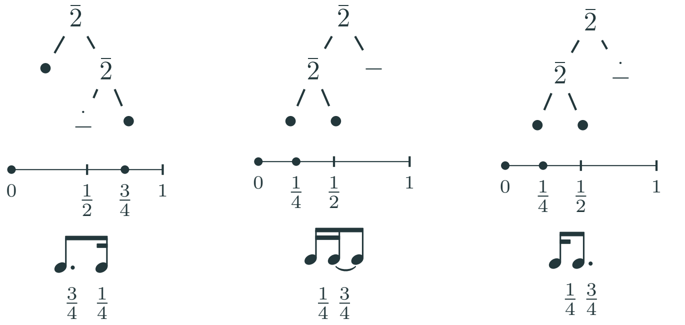
$$\mathrm{dur}(p) \;=\; \frac{\mathrm{dur}(\mathrm{parent}(p))}{\mathrm{arity}(\mathrm{parent}(p))} + \mathrm{dur}\big(\mathrm{nextleaf}(p)\big)$$

otherwise, if $p \neq \mathrm{root}(t)$

$$\mathrm{dur}(p) \;=\; \frac{\mathrm{dur}(\mathrm{parent}(p))}{\mathrm{arity}(\mathrm{parent}(p))}$$

Ties and Dots (2)

One dot augments the duration of a note by $\frac{1}{2}$ of its original duration.

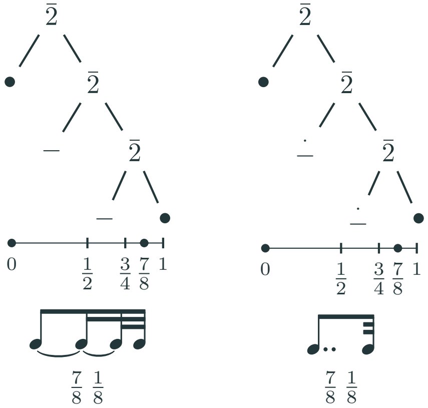


Ties and Dots (3)

Dots can be cumulated, *i.e.* n dots after a note augments its duration by $\frac{2^n-1}{2^n}$ of its original duration ($\frac{3}{4}$ for 2 dots, $\frac{7}{8}$ for 3 dots).

In practice, $n \leq 3$.

Example with 2 dots.



Dots equivalence

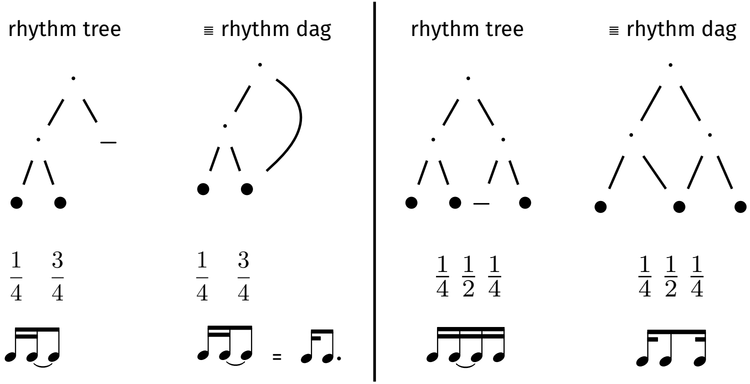
Dots are useful to switch between simple and compound meters.

$\rightarrow$ equational theory to convert one into the other (in a few slides)

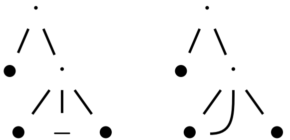
Rhythm DAGs

Rhythm Dags vs Rhythm Trees

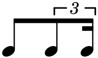
representation of sum of durations by node sharing

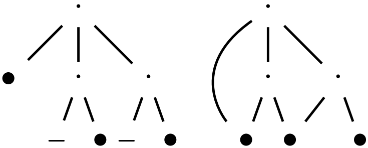


Rhythm Dags vs Rhythm Trees (2)

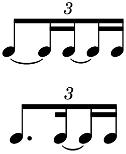


$$\frac{1}{2} \quad \frac{1}{3} \quad \frac{1}{6}$$





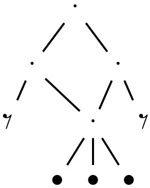
$$\frac{1}{2} \quad \frac{1}{3} \quad \frac{1}{6}$$



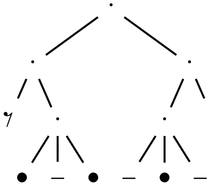
Rhythm Dags vs Rhythm Trees (3) : ratios

representation of a whole bar by a Dag.
both examples contain a join (node sharing) followed by a fork (division)

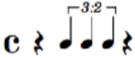
rhythm dag



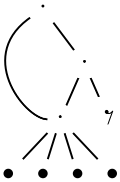
≡ rhythm tree



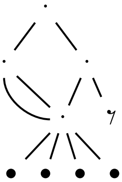
$$\frac{1}{4} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{4}$$



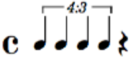
rhythm dag



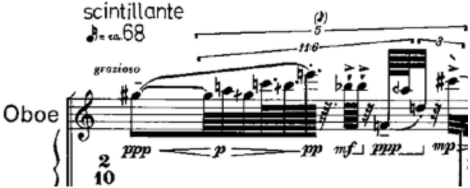
≡ rhythm dag



$$\frac{3}{16} \quad \frac{3}{16} \quad \frac{3}{16} \quad \frac{3}{16} \quad \frac{1}{4}$$



p:q = p in the time of q



Brian Ferneyhough
Etudes Transcendantales (1982-85)
oboe part
first bar of movement 1

Structural Theory of Rhythm Notation

Rewrite Rules

$p = 2, 3 ...$

addition of rests

$$p(\circ, \ldots, \circ) \to \circ \qquad\qquad p(\circ, -, \ldots, -) \to \circ$$

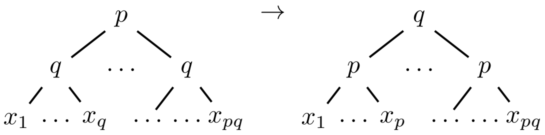
normalization of ties

$$p(-, \ldots, -) \to - \qquad\qquad p(\bullet, -, \ldots, -) \to \bullet$$

(same for dots)

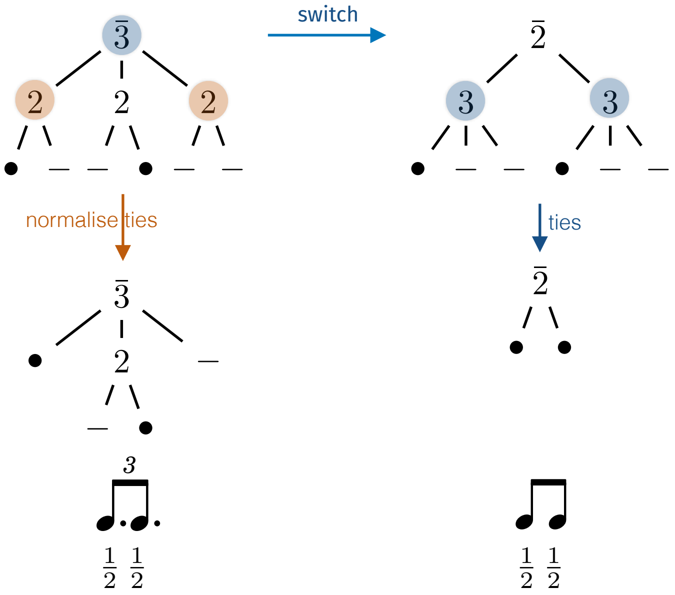
arity switch

$$p\big(q(x_{1,1}, \ldots, x_{1,q}), \ldots, q(x_{p,1}, \ldots, x_{p,q})\big) \to q\big(p(x_{1,1}, \ldots), \ldots, p(\ldots, x_{p,q})\big)$$

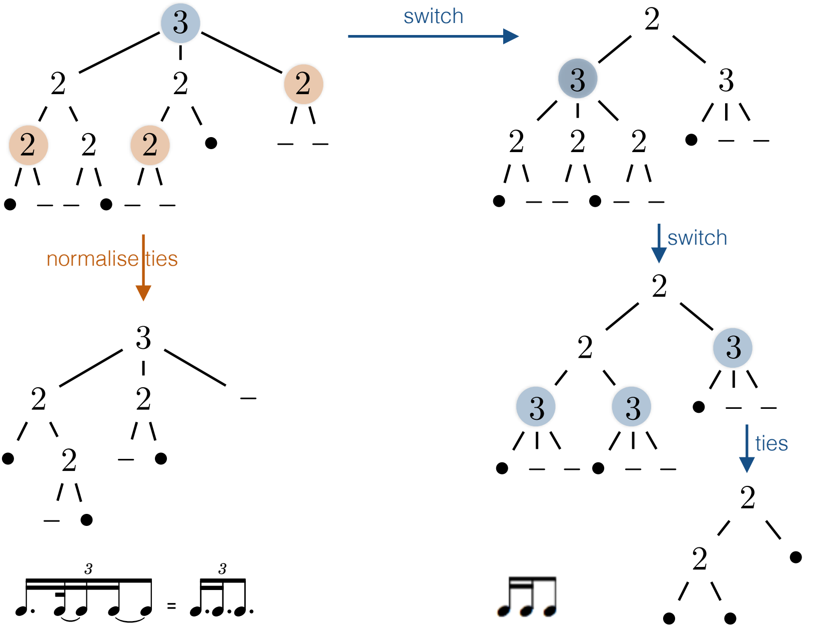


Rewriting Example

Switch from ternary to binary meter.



Rewrite Peak



Equivalence

The rewrite rules preserve the durations of leaves.

Equivalent trees

Every two rhythm trees equivalent modulo $\leftrightarrow$ have the same duration sequence.

Confluence

The TRS STRN is not ground confluent.

Hence there is in general no canonical form for rhythm trees.

But that's actually not needed!

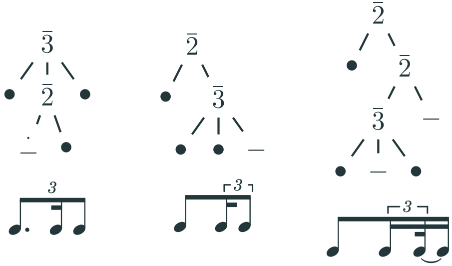
- showing equivalence of two RTs is easy (compute duration sequences)
- generation of trees equivalent to a given tree t is more interesting.

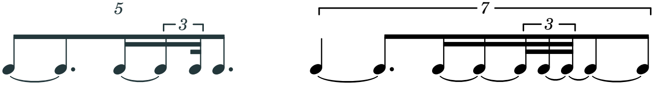
We want efficient representation and enumeration of equivalence classes (sets of rhythm trees with same duration sequence).

Tree Enumeration Example

Example: enumerate Rhythm Trees of $\mathcal{T}(\Sigma)$ with a duration sequence: $\frac{1}{2}\ \frac{1}{6}\ \frac{1}{3}$.

e.g.



and 

. . .

Tree Enumeration (principle)

enumerate Rhythm Trees of $\mathcal{T}(\Sigma)$ with a duration sequence: $\frac{1}{2}\ \frac{1}{6}\ \frac{1}{3}$, by increasing size.

key property: monotonicity of size.

csq: a smallest tree (in size) is made of smallest subtrees.

Tree Enumeration Example (2)

enumerate Rhythm Trees of $\mathcal{T}(\Sigma)$ with a duration sequence: $\frac{1}{2}\ \frac{1}{6}\ \frac{1}{3}$, by increasing size.

$$
\begin{array}{ccc}
 & [\frac{1}{2}, \frac{1}{6}, \frac{1}{3}] & \\
 / & & \backslash \\
[\frac{1}{2}] & & [\frac{1}{6}, \frac{1}{3}]
\end{array}
\qquad
\begin{array}{ccc}
 & [\frac{1}{2}, \frac{1}{6}, \frac{1}{3}] & \\
 / & | & \backslash \\
[\frac{1}{3}] & [-\frac{1}{6}, \frac{1}{6}] & [\frac{1}{3}]
\end{array}
\quad \ldots
$$

$$
\begin{array}{rcl}
best[\frac{1}{2}, \frac{1}{6}, \frac{1}{3}] & = & \min \left(\begin{array}{l} \bar{2}\big(best[\frac{1}{2}], best[\frac{1}{6}, \frac{1}{3}]\big) \\ \bar{3}\big(best[\frac{1}{3}], best[-\frac{1}{6}, \frac{1}{6}], best[\frac{1}{3}]\big) \\ \ldots \end{array} \right) \\
 & = & \min \left(\begin{array}{l} \bar{2}\big(\bullet, best[\frac{1}{6}, \frac{1}{3}]\big) \\ \bar{3}\big(\bullet, best[-\frac{1}{6}, \frac{1}{6}], \bullet\big) \\ \ldots \end{array} \right)
\end{array}
$$

where $\min$ is the tree of minimal size.

Tree Enumeration Example (3)

$$\begin{array}{ccc} & [\frac{1}{6},\frac{1}{3}] & \\ / & & \backslash \\ [\frac{1}{6},\frac{1}{12}] & & [-\frac{1}{4}] \end{array} \qquad \begin{array}{ccc} & [\frac{1}{6},\frac{1}{3}] & \\ / & | & \backslash \\ [\frac{1}{6}] & [\frac{1}{6}] & [-\frac{1}{6}] \end{array} \quad \ldots$$

$$\begin{aligned} best[\tfrac{1}{6},\tfrac{1}{3}] &= \min\begin{pmatrix} \bar{2}\big(best[\frac{1}{6},\frac{1}{12}], best[-\frac{1}{4}]\big) \\ \bar{3}\big(best[\frac{1}{6}], best[\frac{1}{6}], best[-\frac{1}{6}]\big) \\ \ldots \end{pmatrix} \\ &= \min\begin{pmatrix} \bar{2}\big(best[\frac{1}{6},\frac{1}{12}], -\big) \\ \bar{3}(\bullet,\bullet,-) \\ \ldots \end{pmatrix} \end{aligned}$$

By bounding the branching and depth,
computation of $best$ is exponential in time.

Enumeration of 1st best, 2d best, 3d best...
by maintaining set of candidate trees instead of $\min$.

given:

a finite description D of a set L of allowed RTs
a RT t

compute:

a finite description D' of the subset $L' \subseteq L$ of RTs
with the same rhythmic value as t.

enumerate:

the trees in L'.

objective:

- size of D' linear in the size of D.
- enumeration of the k best trees in time $O\big(k.size(D')^2\big)$.

Rhythmic Languages

Music Notation & Tree Series

Is Music a Language?

Leonard Berstein
Norton Lectures at Harvard, 1973
« The Unanswered Question: Six Talks at Harvard »

idea of music as a kind of universal language
notion of a worldwide, « inborn musical grammar »

cf. **Noam Chomsky** « Language and Mind »
theory of innate grammatical competence

Is Music Notation a Language?

Music Notation Processing as a particular case of Natural Language Processing ?

- musical deep structure:
  melodic motives and phrases, chordal progressions, rhythmic figures, etc
- musical surface structure:
  the actual music (sequence of notes)

Music Notation is a Domain Specific Language (not a natural language)
- formal language for exchange (transmission),
- encoding with a small number of symbols,
- semantics (divisions of time).

→ definition of fragments (sub-language of music notation)
preferred in certain contexts.
as a regular tree language.

do prefer notations like this

or that?

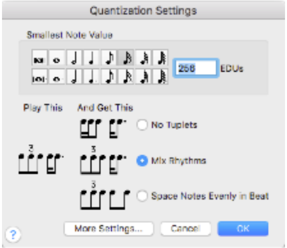
source: Donald Byrd
Indiana University Bloomington.

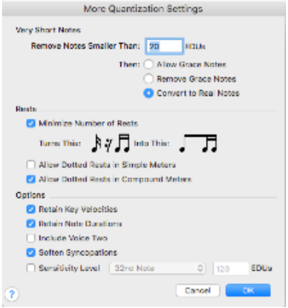
music score editor

Finale quantization Settings
dialog boxes

Mix definition of output rhythm langage
and quantization options

→ Trial & error approach to transcription





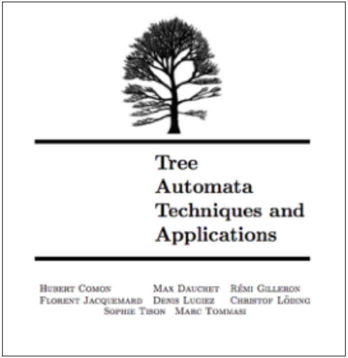
Reading Common Western Music Notation

Common Western Music Notation is a language for Real-Time execution:

- it must be parsable easily, on-the-fly, by performers
- counting symbols (or other computations) cannot be afforded
- it must give clues of the meter (accents)

It is crucial for a music score to be easily readable
→ importance of notation choices and preferences

Tree Automata



http://tata.gforge.inria.fr

TA

A tree automaton $\mathcal{A} = \langle Q, \Delta \rangle$ over a signature Σ is made of

- a finite set of state symbols $Q = \{q, q_0, \ldots\}$ disjoint from Σ,
- a finite set Δ of rewrite rules over $\Sigma \cup Q$ (transitions),
  of the form $a(q_1, \ldots, q_n) \to q_0$
  where $a \in \Sigma$, of arity $n \geq 0$ and $q_0, \ldots, q_n \in Q$.

The language of $\mathcal{A}$ in a state $q \in Q$ is the set of ground terms

$$\mathcal{L}_q = \{t \in \mathcal{T}(\Sigma) \mid t \xrightarrow[\Delta]{*} q\}$$

Acyclic Tree Automaton $\mathcal{A}$ for Rhythm Trees with:

division by 2 and then by 2 or 3,

or division by 3 and then by 2.

$Q = \{q_0, q_1, q_2, q_3\}$

$\Delta =$

$$
\begin{array}{lllllllllllll}
\bar{2}(q_1, q_1) & \to & q_0 & \bar{3}(q_2, q_2, q_2) & \to & q_0 & \bullet & \to & q_0 & & & \\
\bar{2}(q_3, q_3) & \to & q_1 & \bar{3}(q_3, q_3, q_3) & \to & q_1 & \bullet & \to & q_1 & - & \to & q_1 \\
\bar{2}(q_3, q_3) & \to & q_2 & & & & \bullet & \to & q_2 & - & \to & q_2 \\
 & & & & & & \bullet & \to & q_3 & - & \to & q_3
\end{array}
$$

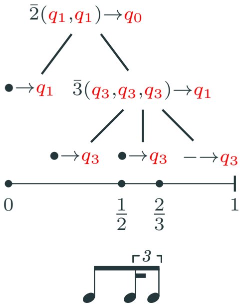
Example RTs in TA language

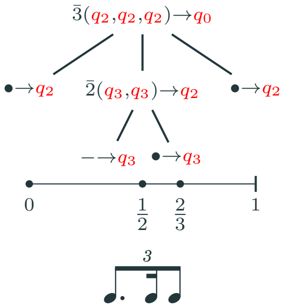
computation of $\mathcal{A}$ on $t = \bar{2}\big(\bullet, \bar{3}(\bullet, \bullet, -)\big)$

$$t \xrightarrow[\Delta]{} \bar{2}\big(q_1, \bar{3}(\bullet, \bullet, -)\big) \xrightarrow[\Delta]{} \bar{2}\big(q_1, \bar{3}(q_3, q_3, q_3)\big) \xrightarrow[\Delta]{} \bar{2}(q_1, q_1) \xrightarrow[\Delta]{} q_0$$

represented by a tree (called run)

- with the shape of t
- labeled by the rules of Δ involved





exercise: when can we label a node by a dot instead of a tie?

we can characterize the following patterns with dots
(remember that a dot must have half the duration the related note):

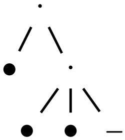
```
         .                .
        / \              / \
       ●   .            .   ∸
          / \          / \
         ∸   x        x   ●
```

using the following production rules:

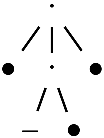
$$
\begin{array}{llll}
q_0 \to q_1, q_2 & q_1 \to \bullet & q_2 \to q_{\cdot}, q_x & q_{\dot{-}} \to \dot{-} \\
q_0 \to q_2', q_1' & q_1' \to \dot{-} & q_2' \to q_x, q_{\bullet} & q_{\bullet} \to \bullet \\
q_x \to \ldots & & &
\end{array}
$$

Introduction of weight values

in some cases, you may prefer division of beat by 2
e.g. (binary meter)



in some other cases, you may prefer division of beat by 3
e.g. (ternary meter)



but we do not want to exclude completely the other case...
→ quantify the preferences in term of beaming *etc.*
→ introduction of weight values in TA transition rules

Weight values are chosen in a semiring

Semirings

A *semiring* $\mathcal{S} = \langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$ is a structure with

- a domain $\mathbb{S} = dom(\mathcal{S})$
- two associative binary operators $\oplus$ and $\otimes$
  with neutral elements $\mathbb{0}$ and $\mathbb{1}$; and such that
- $\oplus$ is commutative
- $\otimes$ distributes over $\oplus$: $\forall x, y, z \in \mathbb{S}$, $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$,
- $\mathbb{0}$ is absorbing for $\otimes$: $\forall x \in \mathbb{S}$, $\mathbb{0} \otimes x = x \otimes \mathbb{0} = \mathbb{0}$

Intuitively,

$\oplus$ is for selection of a best value

$\otimes$ is for composition of values

Semiring Properties

$\mathcal{S}$ is commutative if $\otimes$ is commutative

$\mathcal{S}$ is monotonic *wrt* a partial ordering $\leq$ iff for all x, y, z, $x \leq y$ implies $x \oplus z \leq y \oplus z$, $x \otimes z \leq y \otimes z$ and $z \otimes x \leq z \otimes y$.

$\mathcal{S}$ is idempotent if $\forall x \in \mathcal{S}$, $x \oplus x = x$.

in this case, the natural ordering $\leq_{\mathcal{S}}$ defined by:

$$\forall x, y, x \leq_{\mathcal{S}} y \text{ iff } x \oplus y = x$$

Semirings Examples

semiring	domain	$\oplus$	$\mathbb{0}$	$\otimes$	$\mathbb{1}$	natural ordering
Boolean	$\{0,1\}$	$\vee$	0	$\wedge$	1	$\mathbb{1} \leq_{\mathcal{S}} \mathbb{0}$
Viterbi	$[0,1] \subset \mathbb{R}_+$	max	0	.	1	$x \leq_{\mathcal{S}} y$ iff $x \geq y$
min-plus	$\mathbb{R}_+ \cup \{+\infty\}$	min	$+\infty$	$+$	0	$x \leq_{\mathcal{S}} y$ iff $x \leq y$
max-plus	$\mathbb{R} \cup \{-\infty\}$	max	$-\infty$	$+$	0	$x \leq_{\mathcal{S}} y$ iff $x \geq y$

These semirings are

commutative: $\otimes$ is commutative

idempotent: $\forall x$, $x \oplus x = x$

have an induced total natural ordering $\leq_{\mathcal{S}}$ defined by:

$$\forall x, y, x \leq_{\mathcal{S}} y \text{ iff } x \oplus y = x$$

monotonic *wrt* $\leq_{\mathcal{S}}$: $\forall x, y, z$, $x \leq_{\mathcal{S}} y$ implies

$x \oplus z \leq_{\mathcal{S}} y \oplus z$

$x \otimes z \leq_{\mathcal{S}} y \otimes z$

WTA

A Weighted Tree Automaton (WTA) $\mathcal{A} = \langle Q, \Delta \rangle$ over a signature Σ and a semiring $\mathcal{S} = \langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$ is made of

- a finite set of state symbols $Q = \{q, q_0, \ldots\}$ disjoint from Σ,
- a finite set Δ of weighted rewrite rules over $\Sigma \cup Q$ and $\mathcal{S}$ of the form $a(q_1, \ldots, q_n) \xrightarrow{w} q_0$ where $a \in \Sigma$, of arity $n \geq 0$, $w \in \mathcal{S}$, and $q_0, \ldots, q_n \in Q$.

The tree series defined by $\mathcal{A}$ and state $q \in Q$ is the function

$$\begin{array}{rccl} \mathcal{A}_q : & \mathcal{T}(\Sigma) & \to & \mathcal{S} \\ & t & \mapsto & \displaystyle\bigoplus_{t \xrightarrow[\Delta]{\sigma} q} weight(\sigma) \end{array}$$

where $weight(\sigma)$, the weight of the rewrite sequence σ is the product with $\otimes$ of the rules of Δ involved.

Such tree series is called recognizable.

Example Weighted Tree Automata

Acyclic WTA $\mathcal{A}$ over a min-plus (tropical) Semiring for Rhythm Trees with:

division by 2 and then by 2 or 3,
or division by 3 and then by 2.

$Q = \{q_0, q_1, q_2, q_3\}$

$\Delta =$

$$
\begin{array}{llllllllll}
\bar{2}(q_1, q_1) & \xrightarrow{25} & q_0 & \bar{3}(q_2, q_2, q_2) & \xrightarrow{45} & q_0 & \bullet & \xrightarrow{15} & q_0 & & & \\
\bar{2}(q_3, q_3) & \xrightarrow{20} & q_1 & \bar{3}(q_3, q_3, q_3) & \xrightarrow{70} & q_1 & \bullet & \xrightarrow{10} & q_1 & - & \xrightarrow{25} & q_1 \\
\bar{2}(q_3, q_3) & \xrightarrow{50} & q_2 & & & & \bullet & \xrightarrow{10} & q_2 & - & \xrightarrow{25} & q_2 \\
 & & & & & & \bullet & \xrightarrow{15} & q_3 & - & \xrightarrow{35} & q_3
\end{array}
$$

Since $\mathbb{0}$ is absorbing ($+\infty$ is the above case of min-plus), a rule with weight $\mathbb{0}$ or a missing transition rule are the same thing.

WTA over a Viterbi Semiring:
generalization of PCFG (Probabilistic Context-Free Grammars)

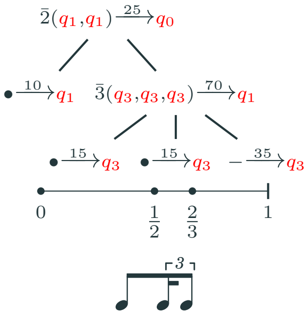
Example RT weight for WTA

the weight $\mathcal{A}_{q_0}(t)$ of $t = \bar{2}\big(\bullet, \bar{3}(\bullet, \bullet, -)\big)$ is the sum with $\oplus$ (min in tropical semiring) of weights of all of its runs headed by q_0.

There is only one run r of $\mathcal{A}$ over t headed by q_0.

$$t \xrightarrow[\Delta]{10} \bar{2}\big(q_1, \bar{3}(\bullet, \bullet, -)\big) \xrightarrow[\Delta]{15+15+35} \bar{2}\big(q_1, \bar{3}(q_3, q_3, q_3)\big) \xrightarrow[\Delta]{70} \bar{2}(q_1, q_1) \xrightarrow[\Delta]{25} q_0$$

Hence $\mathcal{A}_{q_0}(t)$ is the product with $\otimes$ (sum in tropical semiring) of the weights of rules labeling the nodes of r.



$$\mathcal{A}_{q_0}(t) = 170.$$

WTA Determinism

A WTA $\mathcal{A} = \langle Q, \Delta \rangle$ is deterministic if for all $a \in \Sigma$ of arity n and $q_1, \ldots, q_n \in Q$, there exists at most $q \in Q$ such that $a(q_1, \ldots, q_n) \xrightarrow{w \neq \mathbb{0}} q$ is a rule of Δ.

Determinization of WTA : under conditions on semiring $\mathcal{S}$ powerset construction: new states in $\mathcal{S}^Q$, gives a finite state set when $\mathcal{S}$ is locally finite (every finite subset of $\mathcal{S}$ has a finite closure under $\mathbb{0}$, $\mathbb{1}$, $\oplus$, $\otimes$).

Tiburon library [May and Knight 06]

Minimization of deterministic WTA: PTIME for deterministic WTA over commutative semifields [Maletti 09], [Hanneforth, Maletti, Quernheim 17]

1-best

For a WTA $\mathcal{A}$ over Σ and an idempotent semiring $\mathcal{S} = \langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$,
for a state q of $\mathcal{A}$,
find a tree $t \in \mathcal{T}(\Sigma)$ such that $\mathcal{A}_q(t)$ is minimal *wrt* $\leq_{\mathcal{S}}$.

For deterministic WTA, there is a unique run for each tree.

$\rightarrow$ we focus here on the computation of the minimal run.

Hypotheses for the semiring $\mathcal{S}$:

- $\otimes$ is commutative
- $\mathcal{S}$ is monotonic *wrt* $\leq_{\mathcal{S}}$
- $\leq_{\mathcal{S}}$ is total: $\forall x, y$, $x \oplus y = x$ or $x \oplus y = y$

1-best Parsing Example

$$
\begin{array}{lcl lcl lcl lcl}
\bar{2}(q_1, q_1) & \xrightarrow{25} & q_0 & \bar{3}(q_2, q_2, q_2) & \xrightarrow{45} & q_0 & \bullet & \xrightarrow{15} & q_0 & & & \\
\bar{2}(q_3, q_3) & \xrightarrow{20} & q_1 & \bar{3}(q_3, q_3, q_3) & \xrightarrow{70} & q_1 & \bullet & \xrightarrow{10} & q_1 & - & \xrightarrow{25} & q_1 \\
\bar{2}(q_3, q_3) & \xrightarrow{50} & q_2 & & & & \bullet & \xrightarrow{10} & q_2 & - & \xrightarrow{25} & q_2 \\
& & & & & & \bullet & \xrightarrow{15} & q_3 & - & \xrightarrow{35} & q_3
\end{array}
$$

$$
\begin{array}{rcll}
best(q_0) & = & & 25 \otimes best(q_1) \otimes best(q_1), \\
& & \oplus & 45 \otimes best(q_2) \otimes best_1(q_2) \otimes best_1(q_2), \\
& & \oplus & 15 \\
best(q_1) & = & & 20 \otimes best(q_3) \otimes best(q_3), \\
& & \oplus & 70 \otimes best(q_3) \otimes best_1(q_3) \otimes best_1(q_3), \\
& & \oplus & 10 \oplus 25 \\
best(q_2) & = & & 50 \otimes best(q_3) \otimes best(q_3), \\
& & \oplus & 10 \oplus 25 \\
best(q_3) & = & & 15 \oplus 35
\end{array}
$$

1-best Computation for WTAs

1-best

For a WTA $\mathcal{A}$ over Σ and an idempotent semiring $\mathcal{S} = \langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$,
for a state q of $\mathcal{A}$,
find a tree $t \in \mathcal{T}(\Sigma)$ such that $\mathcal{A}_q(t)$ is minimal *wrt* $\leq_{\mathcal{S}}$.

Hypotheses:

$\mathcal{A} = \langle Q, \Delta \rangle$ is acyclic
$\otimes$ is commutative
$\mathcal{S}$ is monotonic *wrt* $\leq_{\mathcal{S}}$
$\leq_{\mathcal{S}}$ is total: $\forall x, y,\ x \oplus y = x$ or $x \oplus y = y$

The following $best(q)$ returns a best run of $\mathcal{A}$ headed by q

$$best(q) = \bigoplus_{\rho_0 = a \xrightarrow{w} q} \rho_0 \oplus \Big[\bigoplus_{\rho = a(q_1,\ldots q_n) \xrightarrow{w} q} \rho\big(best(q_1), \ldots, best(q_n)\big) \Big]$$

Using a table for storing all the $best(q)$,
the time complexity is $O\big(|Q|.\|\Delta\|\big)$.

k-best

For a WTA $\mathcal{A}$ over Σ and an idempotent semiring $\mathcal{S} = \langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$,
for a state q of $\mathcal{A}$, and $k \geq 1$
find the k trees $t \in \mathcal{T}(\Sigma)$ with $\mathcal{A}_q(t)$ is minimal *wrt* $\leq_{\mathcal{S}}$.

Use

- one table for storing all the $best(q, i)$, for $1 \leq i \leq k$, and
- one set of candidate runs $cand(q)$ for each q.

k-best Parsing Example

$$\begin{array}{lcl} \bar{2}(q_1, q_1) & \xrightarrow{25} & q_0 \\ \bar{2}(q_3, q_3) & \xrightarrow{20} & q_1 \\ \bar{2}(q_3, q_3) & \xrightarrow{50} & q_2 \end{array} \quad \begin{array}{lcl} \bar{3}(q_2, q_2, q_2) & \xrightarrow{45} & q_0 \\ \bar{3}(q_3, q_3, q_3) & \xrightarrow{70} & q_1 \end{array} \quad \begin{array}{lclclcl} \bullet & \xrightarrow{15} & q_0 & & & & \\ \bullet & \xrightarrow{10} & q_1 & - & \xrightarrow{25} & q_1 \\ \bullet & \xrightarrow{10} & q_2 & - & \xrightarrow{25} & q_2 \\ \bullet & \xrightarrow{15} & q_3 & - & \xrightarrow{35} & q_3 \end{array}$$

Initially, by monotonicity of $\mathcal{S}$,

$$cand(q_0) = \left\{ \begin{array}{l} 25 \otimes best(q_1, 1) \otimes best(q_1, 1), \\ 45 \otimes best(q_2, 1) \otimes best_1(q_2, 1) \otimes best_1(q_2, 1), \\ 15 \end{array} \right\}$$

Assume that, after computation, we obtain that:

$best(q_0, 1) = 25 \otimes best(q_1, 1) \otimes best(q_1, 1)$

Then start a second round with:

$$cand(q_0) = \left\{ \begin{array}{l} 25 \otimes best(q_1, 1) \otimes best(q_1, 2), \\ 25 \otimes best(q_1, 2) \otimes best(q_1, 1), \\ 45 \otimes best(q_2, 1) \otimes best_1(q_2, 1) \otimes best_1(q_2, 1), \\ 15 \end{array} \right\}$$

...

k-best

For a WTA $\mathcal{A}$ over Σ and an idempotent semiring $\mathcal{S} = \langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$,
for a state q of $\mathcal{A}$, and $k \geq 1$
find the k trees $t \in \mathcal{T}(\Sigma)$ with $\mathcal{A}_q(t)$ is minimal *wrt* $\leq_{\mathcal{S}}$.

Use one table for storing all the $best(q, i)$, for $1 \leq i \leq k$
and one set of candidate runs $cand(q)$ for each q.

Time complexity is $O\big(k.|Q|.\|\Delta\|\big)$.

And after having computed the k bests,
one can continue with the k next (with same complexity).

WTA: Closure Properties

WTA Closure

Let s_1, s_2 be recognizable tree series over Σ and $\mathcal{S}$,
and let $x \in \mathcal{S}$.
The following tree series are recognizable when S commutative:

1. $x \otimes s_1 : \quad t \mapsto x \otimes s_1(t)$ (closure under scalar product)
2. $s_1 \oplus s_2 : \quad t \mapsto s_1(t) \oplus s_2(t)$ (closure under sum)
3. $s_1 \otimes s_2 : \quad t \mapsto s_1(t) \otimes s_2(t)$ (closure under Hadamard product)

1. product by x in the (assumed) final transition rule
2. disjoint union construction.
3. Cartesian product construction.

WTA : closure under scalar product

Assume that the tree series s_1 is recognized by $\mathcal{A}_1 = \langle Q_1, \Delta_1 \rangle$ in a state $p_1 \in Q_1$.

We assume *wlog* that p_1 is not reentering:
$a(q_1, \ldots, q_n) \xrightarrow{\mathbb{0}} q$ as soon as one q_i, at least, is p_1.

Update the weighted transition rules as follows:

$$\begin{array}{rl} a(q_1, \ldots, q_n) \xrightarrow{x \otimes w} p_1 & \text{if } a(q_1, \ldots, q_n) \xrightarrow{w} p_1 \in \Delta_1, \\ a(q_1, \ldots, q_n) \xrightarrow{w} q & \text{if } a(q_1, \ldots, q_n) \xrightarrow{w} q \in \Delta_1 \text{ and } q \neq p_1. \end{array}$$

disjoint union construction

Assume that s_1, s_2 are recognized resp. by

$\mathcal{A}_1 = \langle Q_1, \Delta_1 \rangle$ in a state $p_1 \in Q_1$, and

$\mathcal{A}_2 = \langle Q_2, \Delta_2 \rangle$ in a state $p_2 \in Q_2$

and that Q_1 and Q_2 are disjoint.

We construct a new WTA $\mathcal{A} = (Q_1 \uplus Q_2, \Delta)$.

It recognizes $s_1 \oplus s_2$ in state $\{p_1, p_2\}$ when Δ is defined by:

$$
\begin{array}{lll}
a(q_1, \ldots, q_n) \xrightarrow[\Delta]{w_1} q \text{ if } a(q_1, \ldots, q_n) \xrightarrow{w_1} q \in \Delta_1 & & \text{and } q, q_1, \ldots, q_n \in Q_1 \\
a(q_1, \ldots, q_n) \xrightarrow[\Delta]{w_2} q \text{ if } a(q_1, \ldots, q_n) \xrightarrow{w_2} q \in \Delta_2 & & \text{and } q, q_1, \ldots, q_n \in Q_2 \\
a(q_1, \ldots, q_n) \xrightarrow[\Delta]{\mathbb{0}} q & & \text{otherwise}
\end{array}
$$

WTA : closure under Hadamard product

Cartesian product construction

Assume that s_1, s_2 are recognized resp. by

$\mathcal{A}_1 = (Q_1, \Delta_1)$ in a state $p_1 \in Q_1$, and

$\mathcal{A}_2 = (Q_2, \Delta_2)$ in a state $p_2 \in Q_2$.

and that Q_1 and Q_2 are disjoint.

We construct a new WTA $\mathcal{A} = (Q_1 \times Q_2, \Delta)$.

It recognizes $s_1 \otimes s_2$ in state $\langle p_1, p_2 \rangle$ when Δ is defined by:

$a(\langle q_1^1, q_1^2 \rangle, \ldots, \langle q_n^1, q_n^2 \rangle) \xrightarrow[\Delta]{w_1 \otimes w_2} \langle q^1, q^2 \rangle$
when $a(q_1^1, \ldots, q_n^1) \xrightarrow{w_1} q^1 \in \Delta_1$ and $a(q_1^2, \ldots, q_n^2) \xrightarrow{w_2} q^2 \in \Delta_2$.

Enumeration of Equivalent Classes of Trees

approach similar to the principle used for transcription

Enumeration of equivalent RTs

given:

a WTA $\mathcal{A}$ over Σ and an idempotent semiring $\mathcal{S}$, a state q_0 of $\mathcal{A}$
a RT t

return: a WTA $\mathcal{A}'$ and a state q'_0 of $\mathcal{A}'$ such that for all $t' \in \mathcal{T}(\Sigma)$

$\mathcal{A}'_{q'_0}(t') = \mathcal{A}_{q_0}(t')$ if t' has the same rhythmic value as t,
$\mathcal{A}'_{q'_0}(t') = \mathbb{0}$ otherwise.

enumerate: the tree series $\mathcal{A}'_{q'_0}$, following $\leq_{\mathcal{S}}$.

$\mathcal{A}'$ is the Hadamard product of $\mathcal{A}$ and an automaton $\mathcal{A}_t$ for the divisions of the duration sequence of t (all its rules have weight $\mathbb{1}$).

$q'_0 = \langle q_0, ds(t) \rangle$.

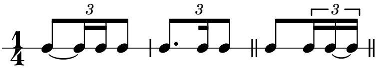
- $\mathcal{A}_t$ can be built in a way that the size of $\mathcal{A}'$ is linear in the size of $\mathcal{A}$.
- the enumeration of the k best trees is done in time $O\big(k.|states(\mathcal{A}')|.\|rules(\mathcal{A}')\|\big)$.

Enumeration of Equivalent RTs (example)

Trees with duration sequence: $\frac{1}{2}\ \frac{1}{6}\ \frac{1}{3}$

$$
\begin{array}{lllllllllll}
\bar{2}(q_1\, q_1) & \xrightarrow{25} & q_0 & \bar{3}(q_2\, q_2\, q_2) & \xrightarrow{45} & q_0 & \bullet & \xrightarrow{15} & q_0 & & \\
\bar{2}(q_3\, q_3) & \xrightarrow{20} & q_1 & \bar{3}(q_3\, q_3\, q_3) & \xrightarrow{70} & q_1 & \bullet & \xrightarrow{10} & q_1 & - \xrightarrow{25} & q_1 \\
\bar{2}(q_3\, q_3) & \xrightarrow{50} & q_2 & & & & \bullet & \xrightarrow{10} & q_2 & - \xrightarrow{25} & q_2 \\
& & & & & & \bullet & \xrightarrow{15} & q_3 & - \xrightarrow{35} & q_3
\end{array}
$$

Only 3 trees (others have weight $\mathbb{0}$).



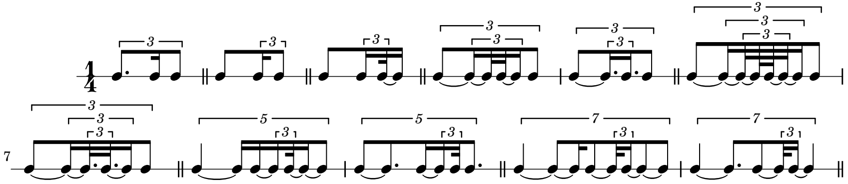
Enumeration of Equivalent RTs (example 2)

$$\bullet \xrightarrow{0.1} q_0$$
$$\bar{2}(q_1, q_1) \xrightarrow{0.25} q_0$$
$$\bar{3}(q_1, q_1, q_1) \xrightarrow{0.45} q_0$$
$$\bar{5}(q_4, q_4, q_4, q_4, q_4) \xrightarrow{0.45} q_0$$
$$\bar{7}(q_4, q_4, q_4, q_4, q_4, q_4, q_4) \xrightarrow{0.45} q_0$$

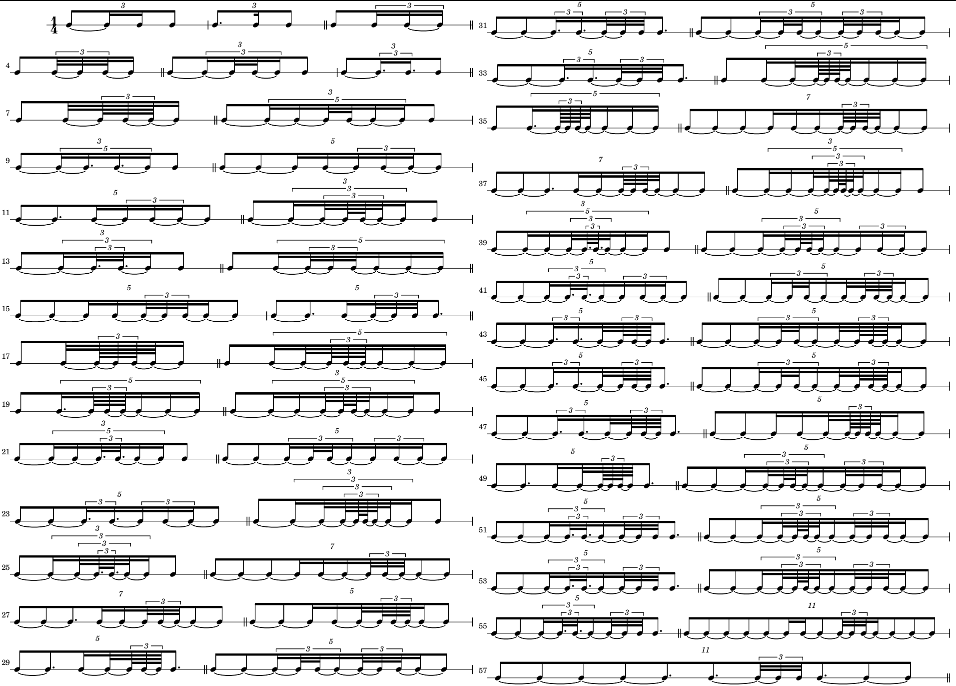
$$\bullet \xrightarrow{0.1} q_1$$
$$- \xrightarrow{0.25} q_1$$
$$\bar{2}(q_2, q_2) \xrightarrow{0.2} q_1$$
$$\bar{3}(q_2, q_2, q_2) \xrightarrow{0.7} q_1$$

$$\bullet \xrightarrow{0.1} q_2$$
$$- \xrightarrow{0.25} q_2$$
$$\bar{2}(q_3, q_3) \xrightarrow{0.5} q_2$$
$$\bar{3}(q_3, q_3, q_3) \xrightarrow{0.5} q_2$$
$$\bullet \xrightarrow{0.15} q_3$$
$$- \xrightarrow{0.35} q_3$$
$$\bar{2}(q_5, q_5) \xrightarrow{0.5} q_3$$

$$\bullet \xrightarrow{0.1} q_4$$
$$- \xrightarrow{0.25} q_4$$
$$\bar{2}(q_5, q_5) \xrightarrow{0.5} q_4$$
$$\bar{3}(q_5, q_5, q_5) \xrightarrow{0.5} q_4$$
$$\bullet \xrightarrow{0.1} q_5$$
$$- \xrightarrow{0.25} q_5$$



Enumeration of More Equivalent RTs (example 3)



Representation of polyrhythms

Chopin Nocturne si majeur opus 9 No 3

merging hands for 1st half of bar 9: [1/5 2/15 1/15 1/5 1/15 2/15 1/5] in 6 alt. notations

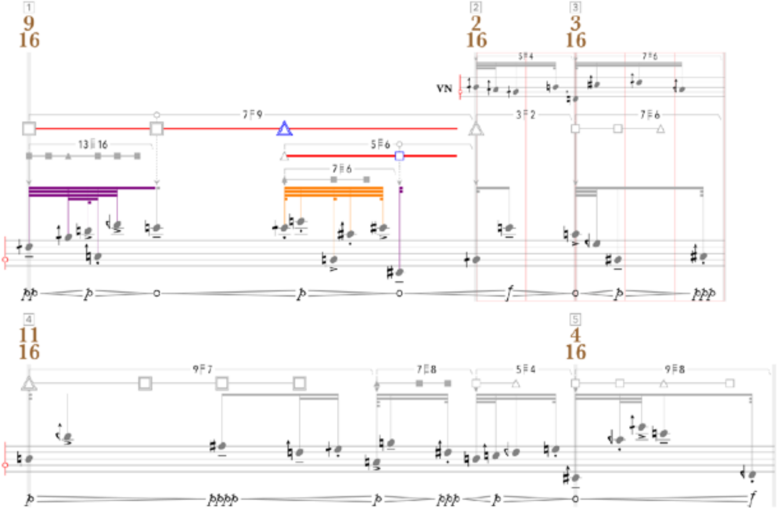
James Bean - dn-m

iPad app that allows performers to interact with the musical notation

simple and clear textual input language for the composer to input his/her music

→ integration of the rhythm enumeration algorithm.

www.jamesbean.info/denm



Automated Music Transcription

Automated Transcription

in Natural Language Processing:
speech-to-text or *voice recognition*

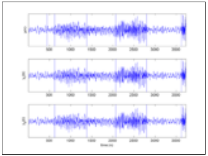
poor's man music transcription

« My dad accidentally texted me with voice recognition while playing the tuba »

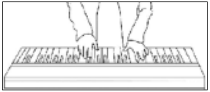
Automated Music Transcription

conversion of a recorded music performance into a music score

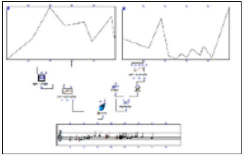
Audio
recording



MIDI Captation
(score editor)



Algorithmic composition
Environment



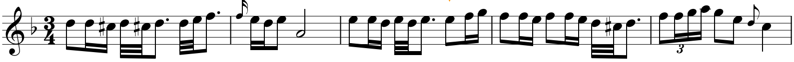
audio MIR

- fund. freq. estimation
- onset detection
- beat tracking ...

piano roll (MIDI file)
- unquantized onsets
- quantized pitches

symbolic MIR

rhythm quantization
& score engraving



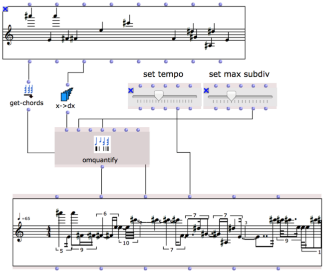
music
score
(XML file)

Grid-based approach to Rhythm quantization

Implemented in most of the Digital Audio Workstation software

assume a fixed minimal duration = tatum.
alignement of dates to closest multiples of tatum

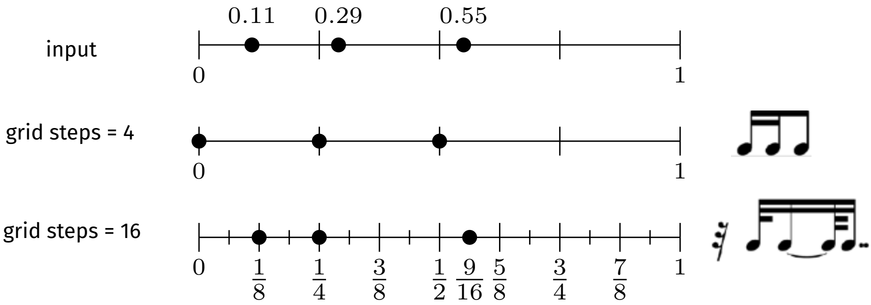
+ efficient quantization: try a few tatum values
– transcription result can be über-complicated



transcription in Open Music

Grid-based Approaches to Rhythm Quantization

overfitting



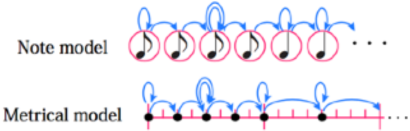
→ find good compromise between precision and complexity of notation (multicriteria optimisation)

→ quantitative parsing

Sequential models of music notation

Can be learned (from score corpora). Popular for transcription since 2000's.

Sequential score models:
HMMs: the probability of a note's duration depends on
the previous note's duration and the input duration



Markov model of note values
[Sagayama et al 2002]

Markov process on meter positions
[Raphael 2001], [Goto et al 2003],
[Cemgil et al 2003]

Hierarchical score models:
Probabilistic Context-Free Grammars (PCFG): [Tanji, Ando, Iba 2008]
model defines the probability of subdivisions (recursively)

+ larger search window (approx. 1 or 2 measure)

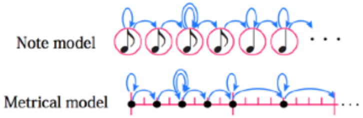
MIDI to score transcription with independent subtasks

transcription approaches with sequential models of durations

1. Rhythm Quantization

unquantized MIDI	→	quantized MIDI
sequential data	→	sequential data

with HMM



2. Score Engraving

quantized MIDI	→	XML score file
sequential data	→	1/2 structured data

delegated to functionality of a score editor (*MIDI import*)

3. Interface between the 2 subtasks? Problems with
   - complex rhythm, deep nesting
   - mixed tuplets
   - rests, grace notes...

unquantized	→	tree	~	XML score file
sequential data	→	1/2 structured data		1/2 structured data

transcription approach with:

- Rhythm Tree representations
- Weighted WTA model of music notation
- computing solutions with
  quantitative parsing techniques (1-best or I*k*-best)
  efficient and modular

a priori Language of Notation

a WTA is given, that represents notational preferences.

example over a min-plus semiring: weight values are penalties (costs)
(toy) language of sequences of 1/4 measures containing RTs

state symbols: q (measure seq.), q_0 (1 measure = 1 beat), q_1, q_2,...

$\top(q_0, q) \xrightarrow{0} q$	$\top(q_0, q_0) \xrightarrow{0} q$		
$\bar{2}(q_1\, q_2) \xrightarrow{6} q_0$	$\bar{3}(q_1\, q_2\, q_2) \xrightarrow{12} q_0$		
$- \xrightarrow{15} q_0$	$\bullet \xrightarrow{7} q_0$	$\bullet_1 \xrightarrow{79} q_0$	$\bullet_2 \xrightarrow{102} q_0$
$- \xrightarrow{2} q_1$	$\bullet \xrightarrow{1} q_1$	$\bullet_1 \xrightarrow{25} q_1$	$\bullet_2 \xrightarrow{64} q_1$
$\bar{2}(q_3, q_3) \xrightarrow{10} q_2$	$\bullet \xrightarrow{2} q_2$		
$\bar{2}(q_4, q_4) \xrightarrow{11} q_3$	$- \xrightarrow{4} q_3$	$\bullet \xrightarrow{1} q_3$	$\bullet \xrightarrow{1} q_4$

a priori Language of Notation

weight = notational complexity

	9	17	41

= product with $\otimes$ of weights of rules involved

run

$$\bar{2}(q_1, q_2) \xrightarrow{6} q_0$$

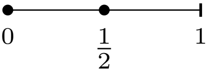
$$\bullet \xrightarrow{1} q_1 \qquad \bullet \xrightarrow{2} q_2$$

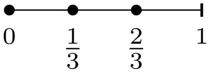
$$\bar{3}(q_1, q_2, q_2) \xrightarrow{12} q_0$$

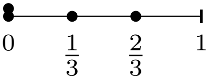
$$\bullet \xrightarrow{1} q_1 \qquad \bullet \xrightarrow{2} q_2 \qquad \bullet \xrightarrow{2} q_2$$

$$\bar{3}(q_1, q_2, q_2) \xrightarrow{12} q_0$$

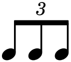
$$\bullet_1 \xrightarrow{25} q_1 \qquad \bullet \xrightarrow{2} q_2 \qquad \bullet \xrightarrow{2} q_2$$





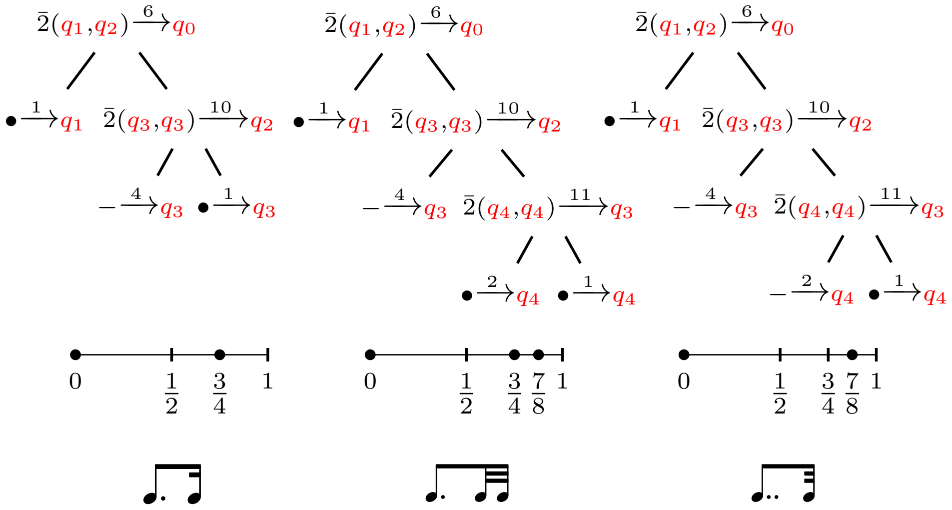


corresponding notation

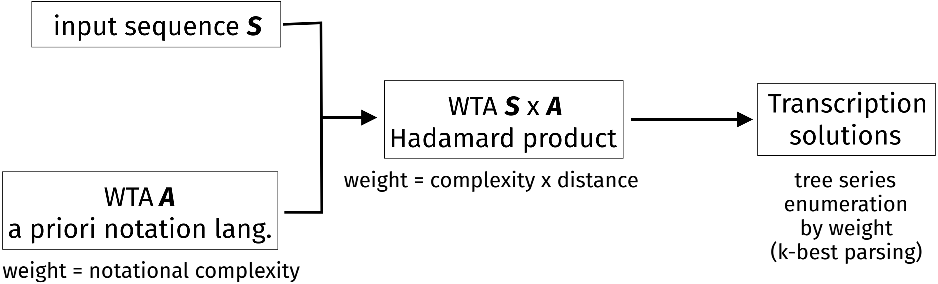


a priori Language of Notation

notational complexity 22 35 35



Rhythm Transcription in by k-best parsing



Transcription by k-best parsing

an input to transcribe is a sequence σ of musical events with dates

it is composed with the a priori WTA $\mathcal{A}$ for notational preferences (Hadamar product).

in the product automaton $\mathcal{A}_\sigma$, the weights in transition are computed by product with $\otimes$ of the notational complexity (defined by $\mathcal{A}$) and a distance of transcription to the input σ.

- with a Viterbi semiring, $\otimes$ is a probability product to maximize.
- with a min-plus semiring $\otimes$ is a sum to minimize.
  similar to scalarization by weighted sum in multi-criteria optimization.

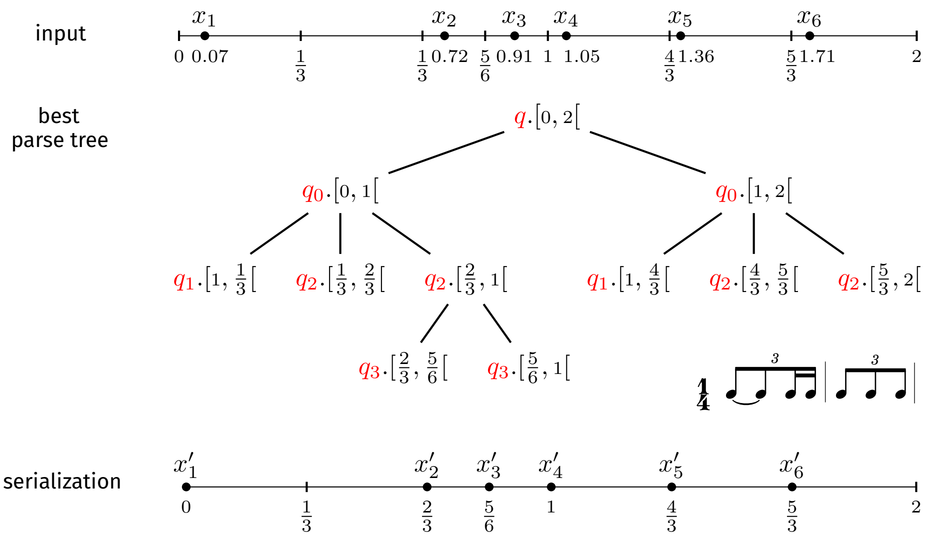
Solution to transcription are computed (as RT) by 1-best or k-best parsing, applied to the product WTA.

for efficiency, the product automaton $\mathcal{A}_\sigma$ is actually computed on-the-fly (lazily) during parsing.

The RT solutions are converted into music scores (XML/MEI).

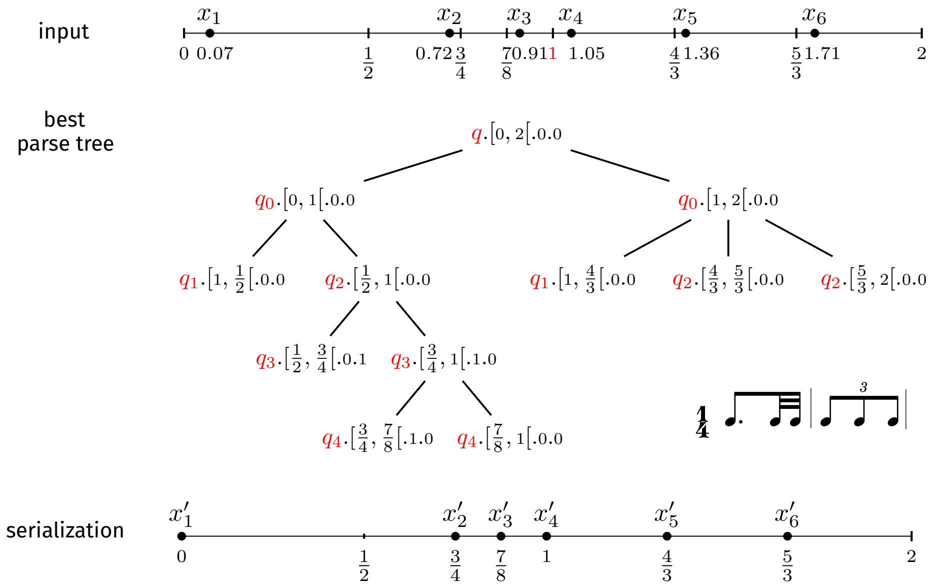
Transcription by 1-best parsing

ex.1: steady tempo, all-left alignments



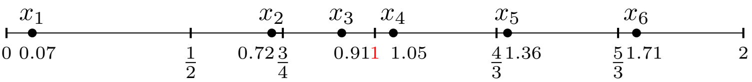
Transcription by 1-best parsing

ex.2: another extension for transcription with steady tempo, alignments to left or right



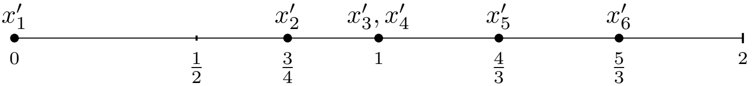
Transcription by 1-best parsing

input



if we reduce the penalty for grace-notes, $q_1 \xrightarrow{7} \bullet_1$
the best parse tree corresponds to:

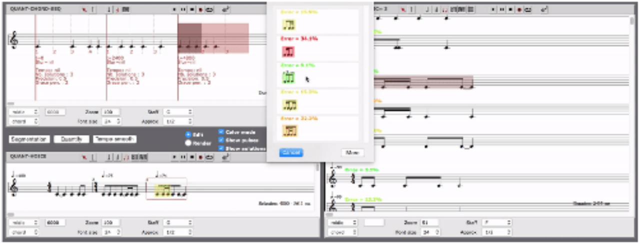
serialization



Former Development : Open Music RQ lib

OpenMusic RQ 2016-17
Adrien Ycart
Jean Bresson

https://forge.ircam.fr/p/omlibraries/downloads/
http://repmus.ircam.fr/cao/rq



CommonLisp/CLOS, 350 functions, 4900 lines of code
UI: Open Music object
    input from chord-seq (notes, onset, dur) + segmentation marks,
    output to voice (OM rhythm trees)
kernel: table construction and enumeration algo

Current Development : sparse C++ library

qparse standalone lib

C++, 25000 lines of code
automata constructions and parsing
MIDI input,
XML/MEI, Lilypond, Guido output
command line prototypes for evaluation
objective: plugin for integration into a score editor

https://qparse.gitlabpages.inria.fr
https://gitlab.inria.fr/qparse/qparselib

transcription: MIDI recording to XML/MEI

https://qparse.gitlabpages.inria.fr
https://gitlab.inria.fr/qparse/qparselib

original score

Beethoven, Trio for violin, cello and piano, op.70 n.2 (2d mov)

transcription of MIDI recording with qparse

transcription: MIDI recording to MusicXML

finale.
music notation software

original score

Beethoven, Trio for violin, cello and piano, op.70 n.2 (2d mov)

transcription of MIDI recording with Finale.

options:
mixed rhythms,
tuplets
smallest note = 32nd
The time signature and the tempo are given.

C++ library. MIDI input, XML/MEI or Lilypond output

https://qparse.gitlabpages.inria.fr
https://gitlab.inria.fr/qparse/qparselib

original score

Polonaise in D minor from Notebook for Anna Magdalena Bach   BWV Anh II 128

transcription of MIDI recording (100 ms) by qparse with generic grammar

MEI output, display with Verovio