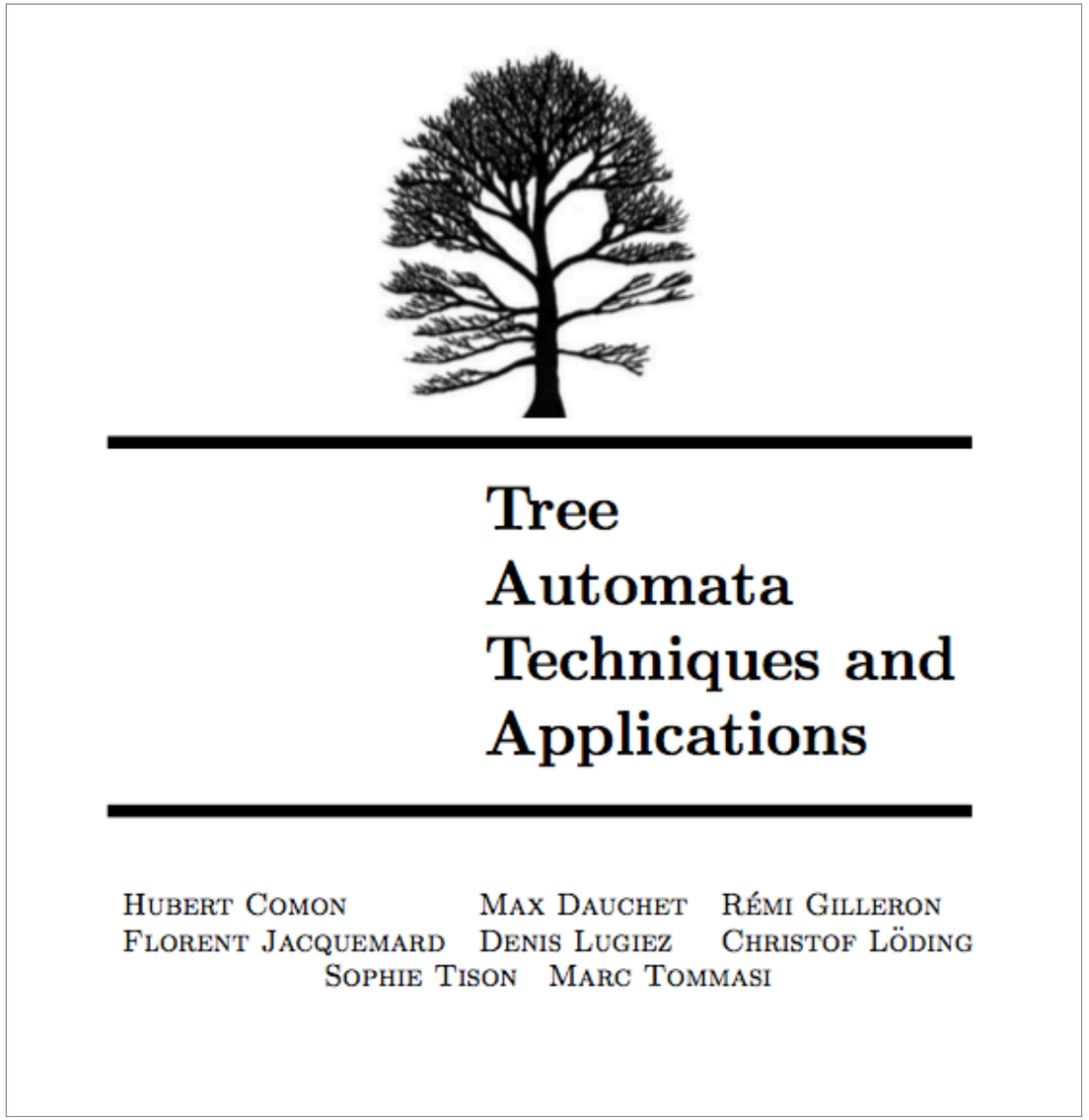
L’age d’or de TATA



Hubert Comon Retirement workshop - June 16 2022 - Saclay

For:

$$x + 0 \to x$$
$$x + s(y) \to s(x + y)$$

the normal forms are: $0, s(0), s(s(0)), \ldots$

This set is generated by the following Regular Tree Grammar with one non-terminal q :

$$q \to 0 \quad q \to s(q)$$

In general, it is the case for sets of rewrite rules with a *linear* left-hand-side (no repetition of variable).
- Normal Forms are the terms that do not embed a left-hand-side,
- Regular Tree Grammar (or equivalently Tree Automata) can do linear pattern matching,
- there languages are closed by complementation.

This is no more true for rules with *non-linear* left-hand-side such as

$$(y \cdot x)/x = y$$

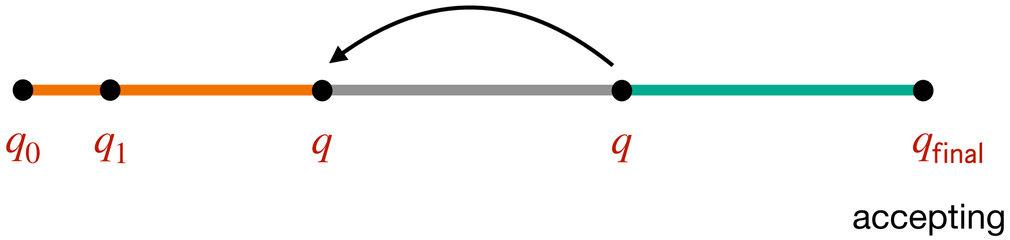
or

$$ins\big(x, ins(x, y)\big) = ins(x, y).$$

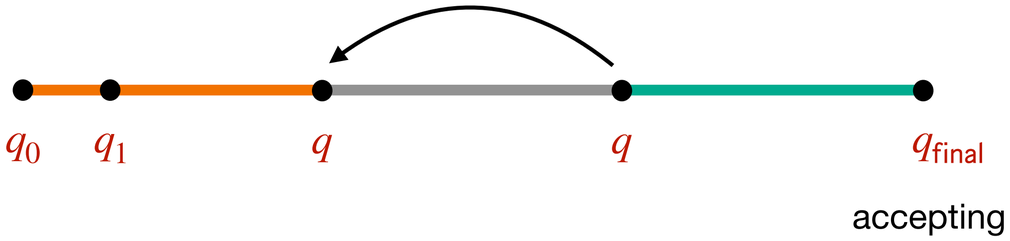
For such cases, tree automata were extended (in Lille) with some *constraints*.
With some local tests of *disequalities* between subtrees during tree automata computation,
one can characterise sets of normal form of ay set of rewrite rules.

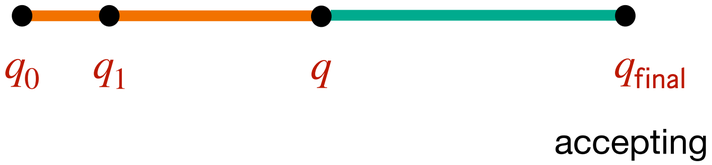
For instance, for the above rule, when detecting a pattern $(y \cdot x)/x'$, the automaton must check that $x \neq x'$.

NFA's run

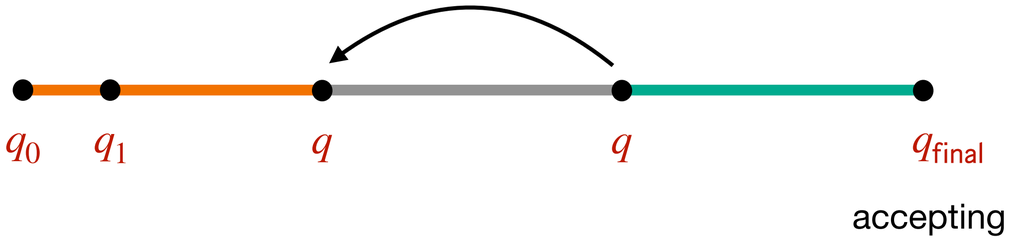


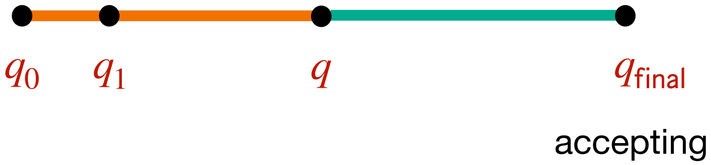
NFA's run



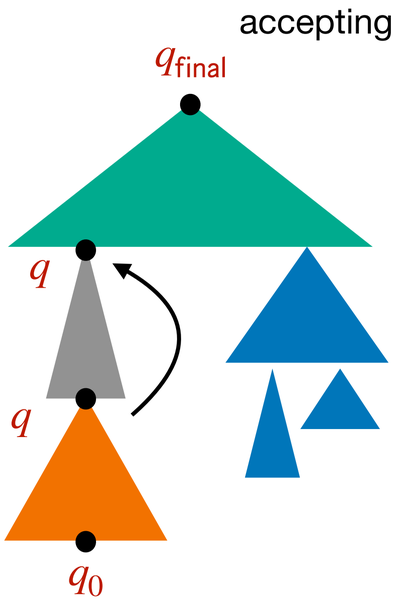


NFA's run

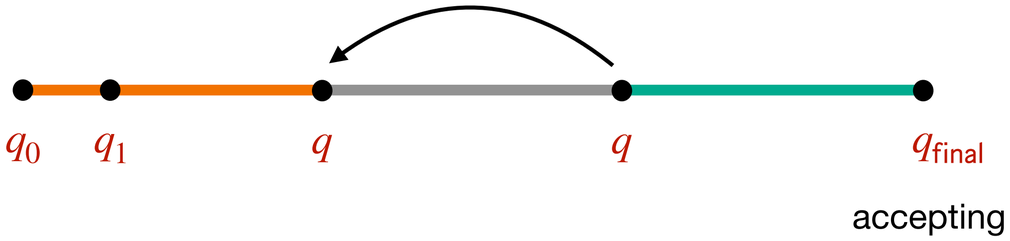


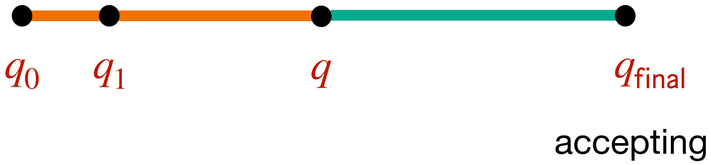


Tree Automaton's run

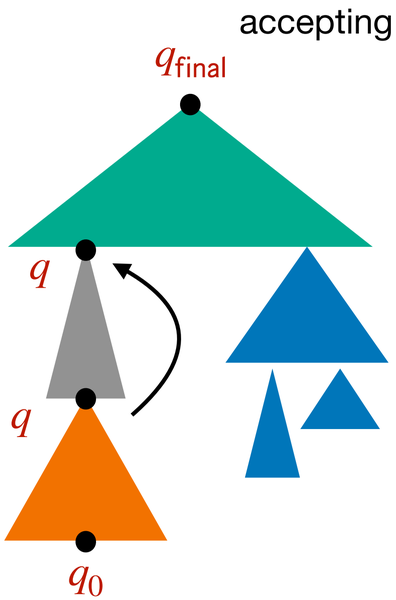


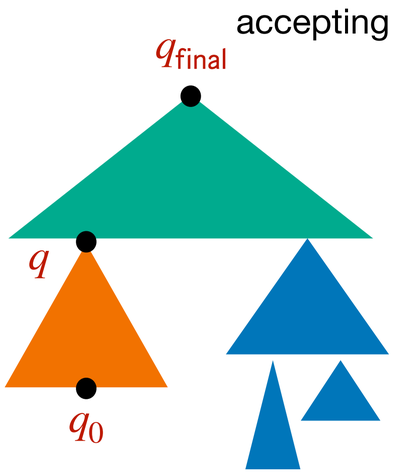
NFA's run



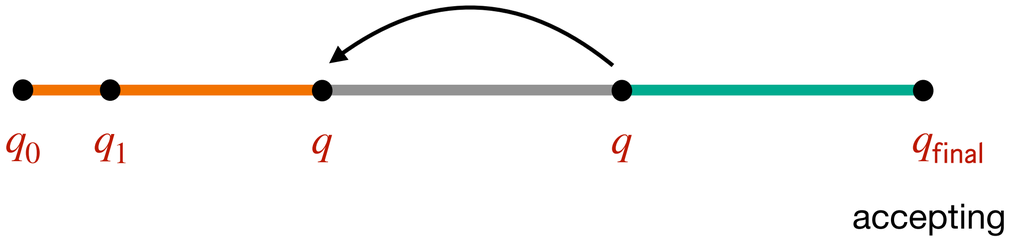


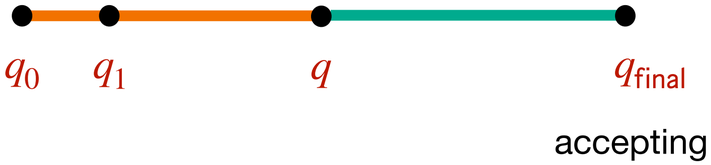
Tree Automaton's run



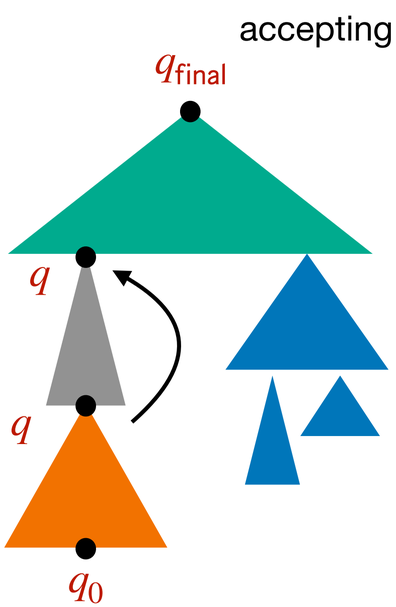


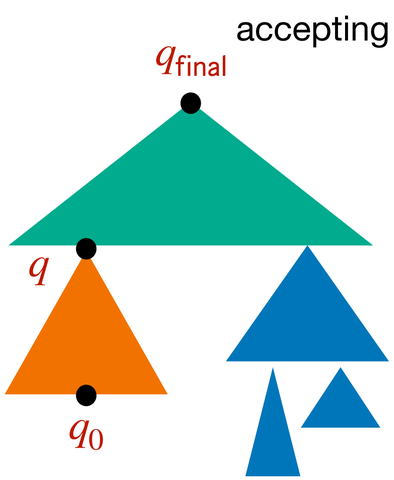
NFA's run

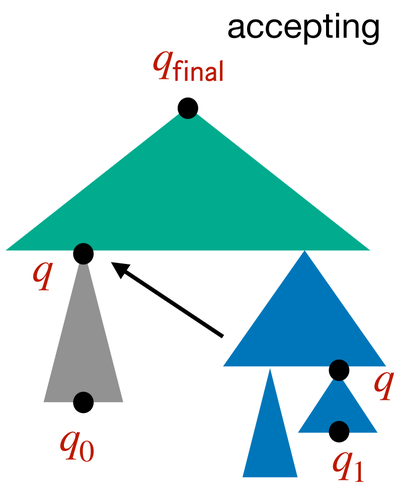




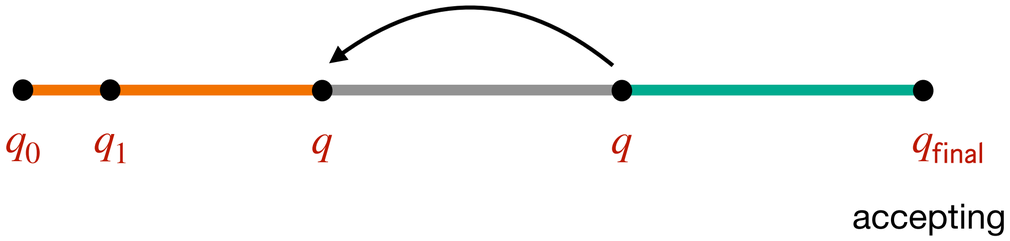
Tree Automaton's run

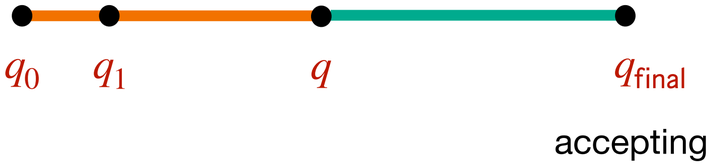




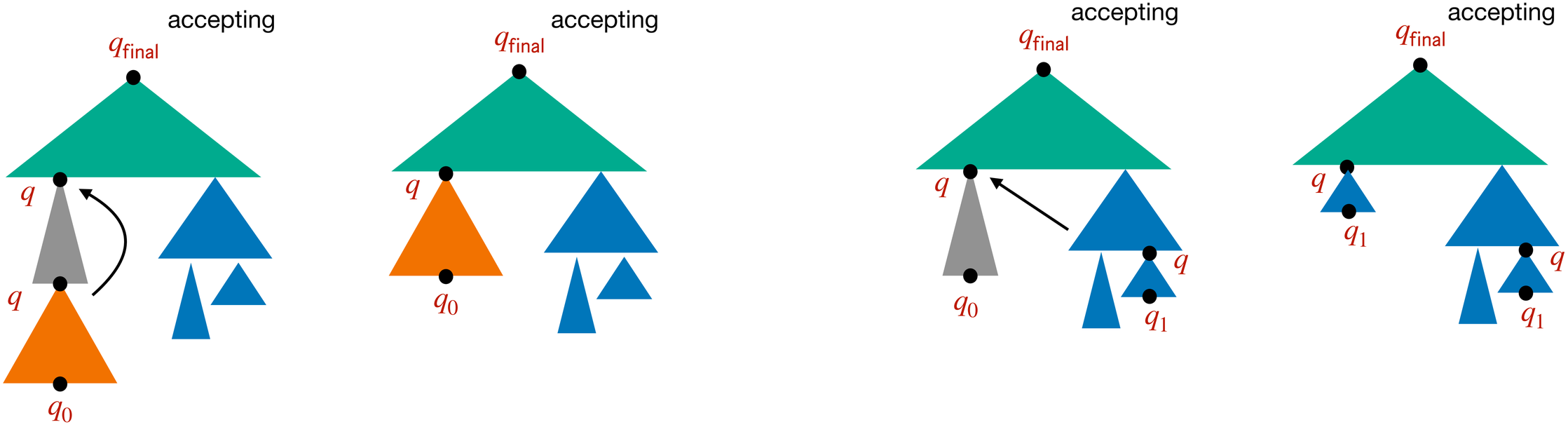


NFA's run

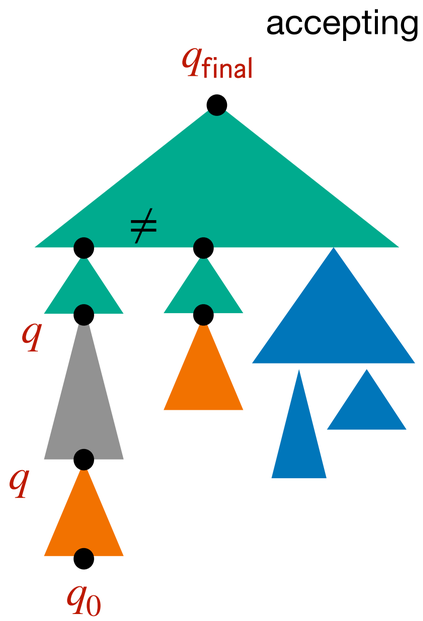




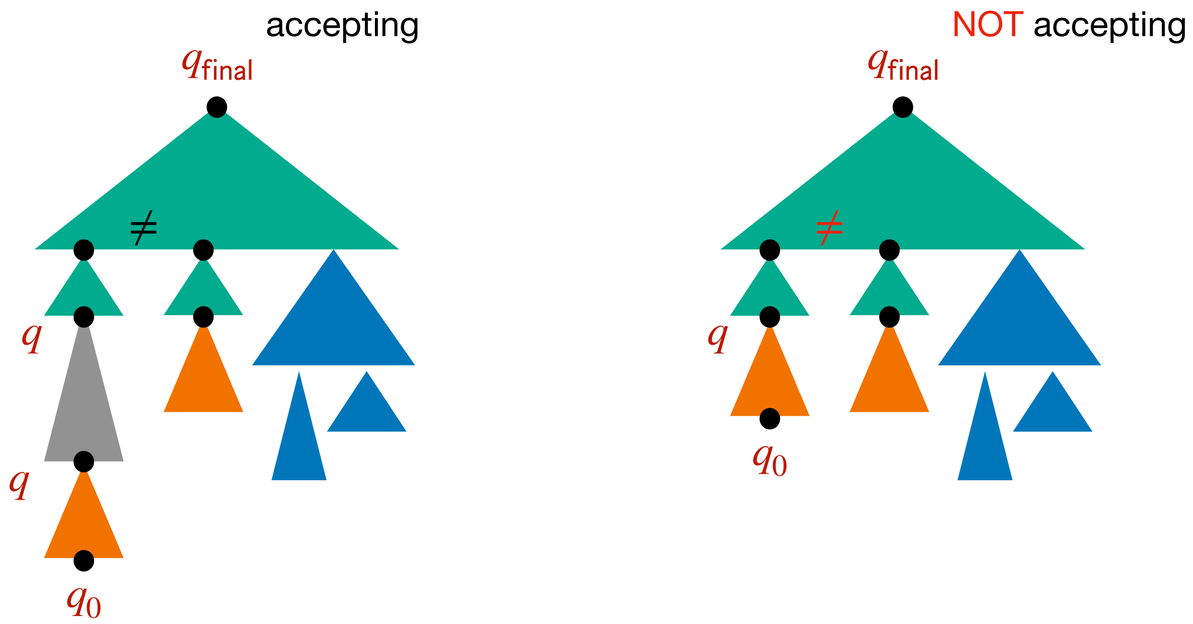
Tree Automaton's run



Constrained Tree Automaton's run

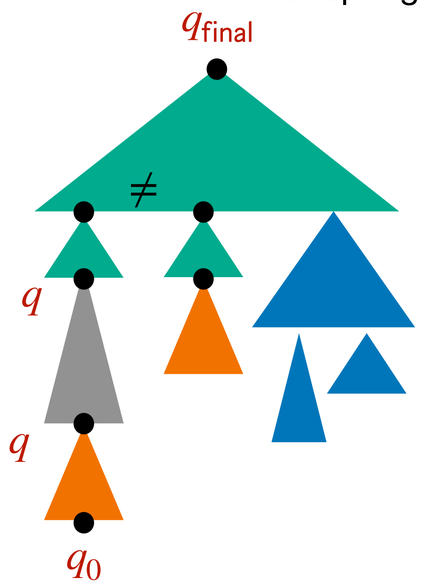


Constrained Tree Automaton's run

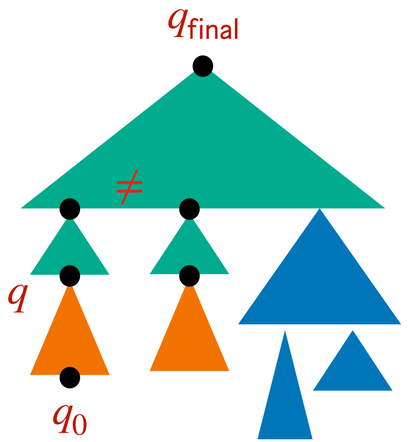


Constrained Tree Automaton's run

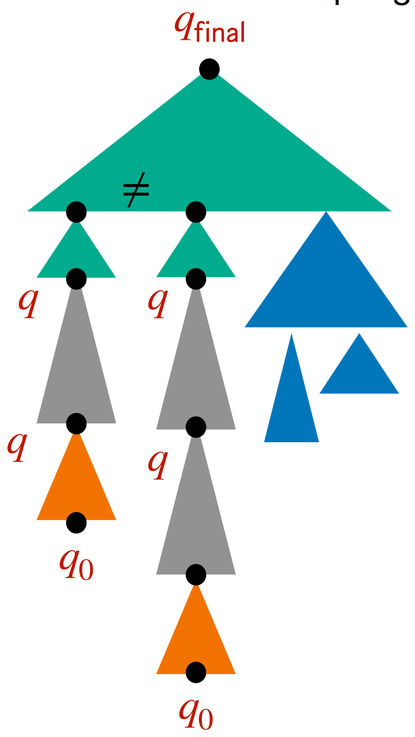
accepting

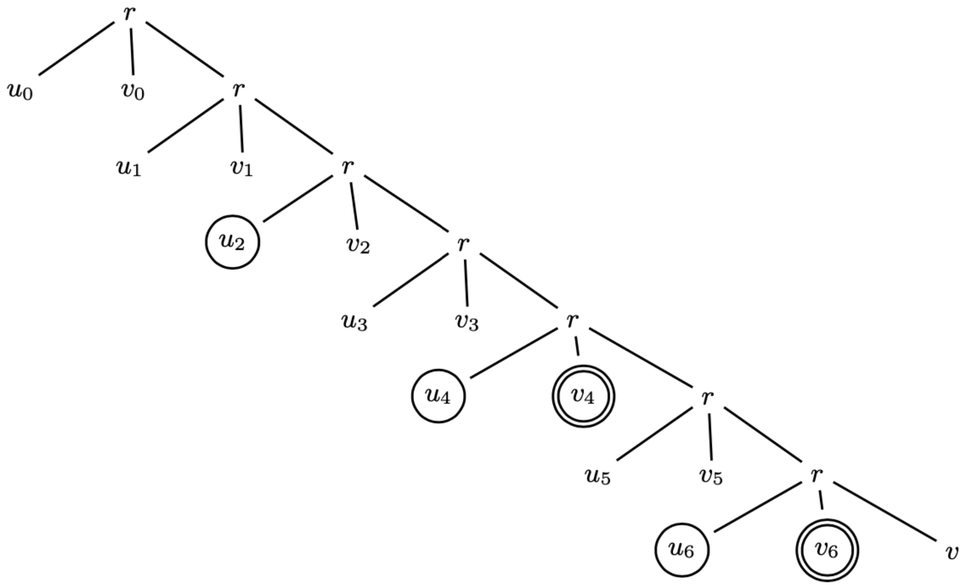


NOT accepting



accepting





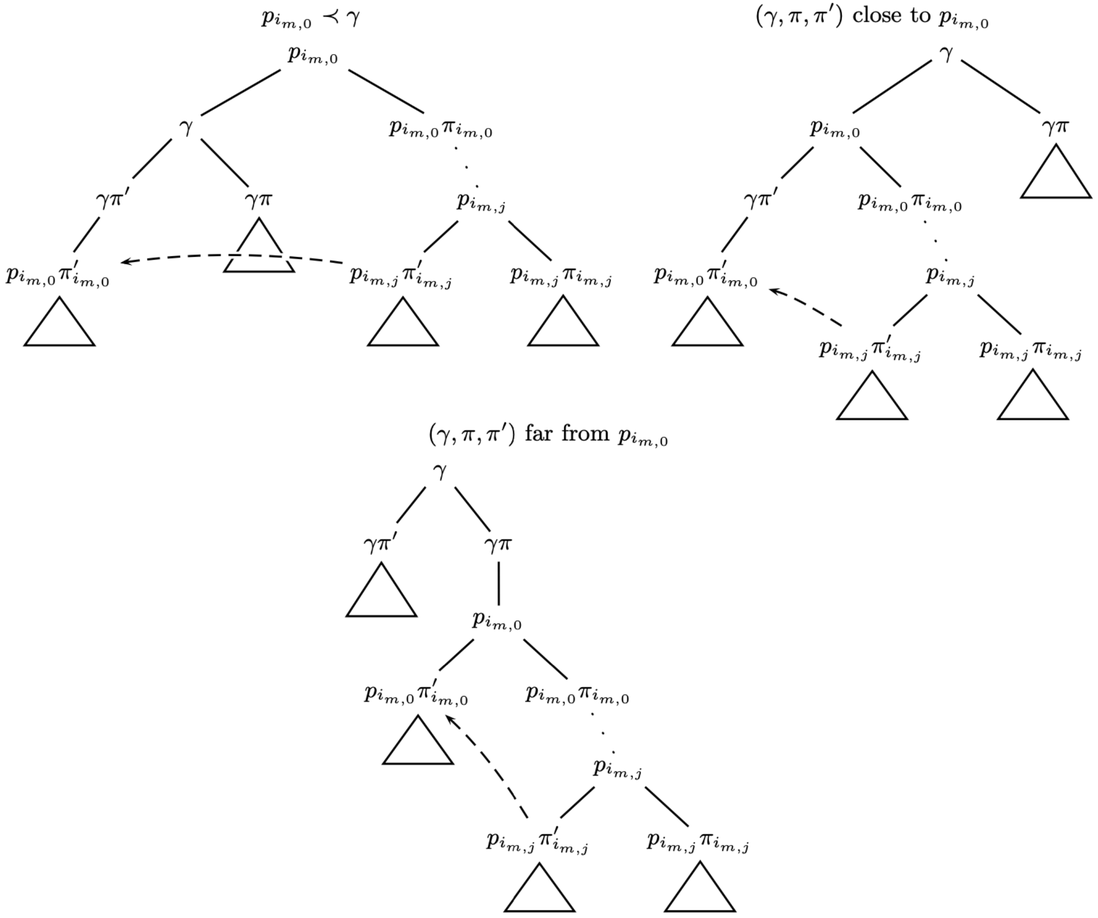
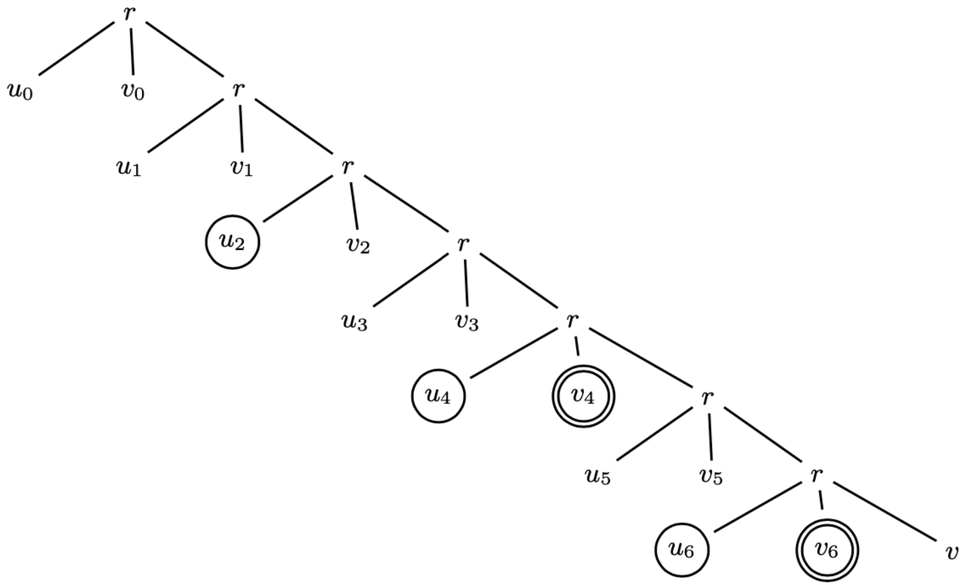


Fig. 8. (γ, π, π') far from $p_{i_{m,0}}\pi'_{i_{m,0}}$.

Ground Reducibility is EXPTIME-complete



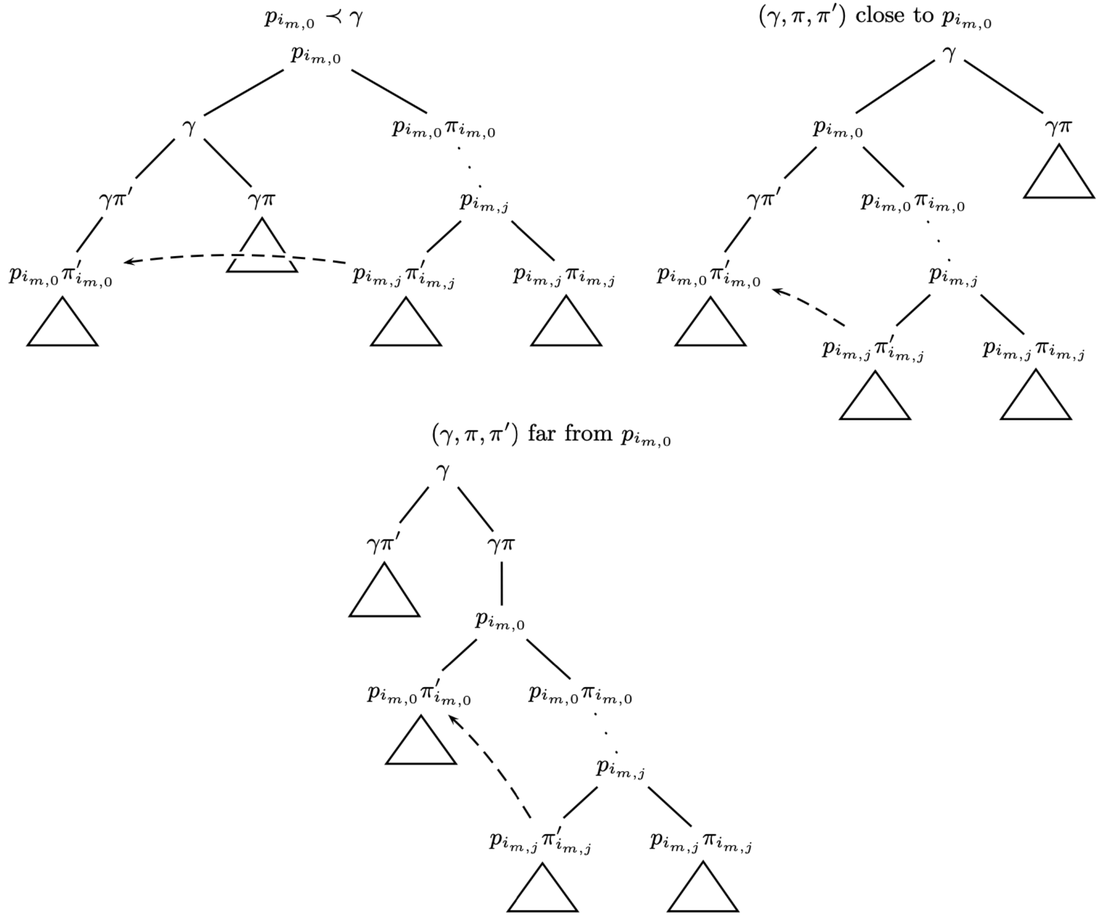
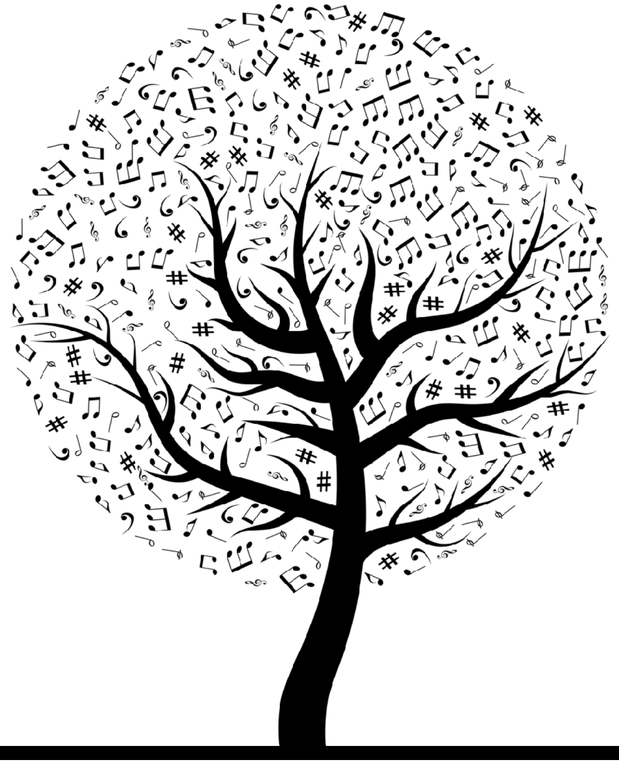


Fig. 8. (γ, π, π') far from $p_{i_{m,0}}\pi'_{i_{m,0}}$.

encore vos lemmes de pompage à la *mord-moi-lnœud* !?



TATA
à la musique

Philippe Rigaux
CNAM

Florent Jacquemard
Inria

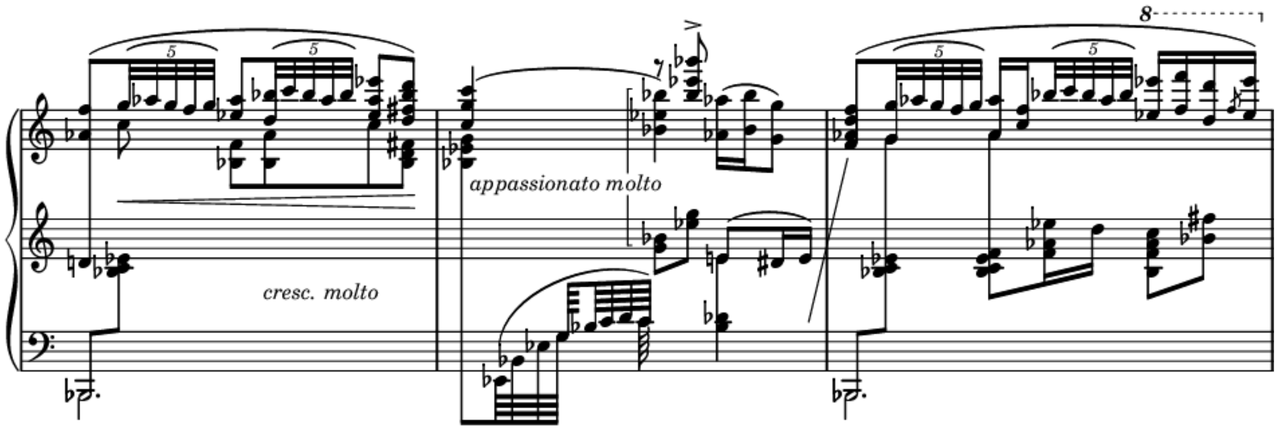
Raphaël Fournier-S’niehotta
CNAM

Lydia Rodriguez-de la Nava
PhD (Codex, Inria)

Tiange Zhu
PhD (Polifonia, EU)

post-doc (Collabscore, ANR)

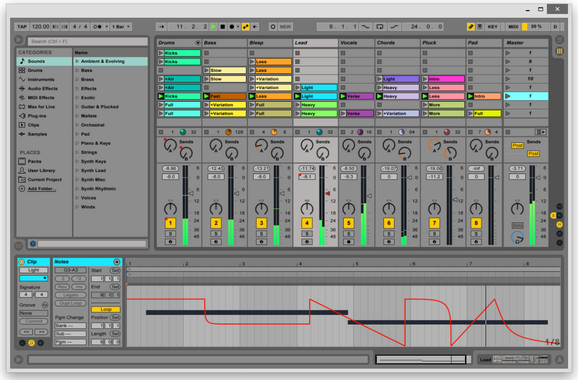
Music Notation Processing

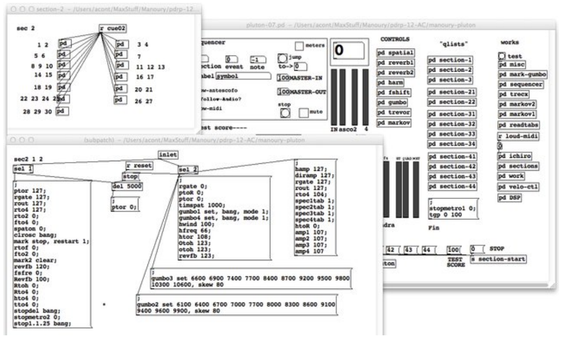


E. Granados’ Goyescas
typesetted with Lilypond

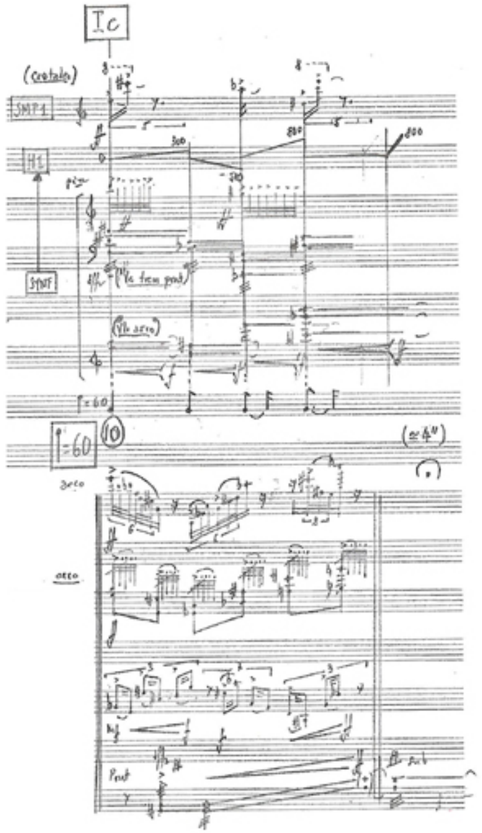
Why studying Music Notation Processing?

Western Music Notation = graphical format for music practice,
in use since ~1000 years (Guido d'Arezzo)





vs



Philippe Manoury
Tensio for string quartet and electronics

(digital) music scores, a natural language for

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

Philippe Rigaux
CNAM

Florent Jacquemard
Inria

Raphaël Fournier-S'niehotta
CNAM

Lydia Rodriguez-de la Nava
PhD (Codex, Inria)

Tiange Zhu
PhD (Polifonia, EU)

post-doc (Collabscore, ANR)

Music Notation Processing

- Structured music scores models
- Music scores languages
  vas-y TATA!
- Search and retrieval
- Similarity metrics
  string and tree edit-distances

Applications

- Databases of digital music scores
  Cultural heritage preservation H2020 Polifonia
- Computational Musicology
  neuma.huma-num.fr - IReMus UMR 8223
- Optical Music Recognition, Crowdsourced correction
  ANR Collabscore - IRISA, BnF, Royaumont
- Automated Music Transcription
  JSPS 採譜 JAIST & Nagoya U. grand Yamaha Music Foundation.

Conversion of a recorded music performance into a music score ~ *speech-to-text* in NLP
a holy grail in Computer Music since 1970's

646 Nature Vol. 263 October 21 1976

articles

Perception of melodies

H. C. Longuet-Higgins

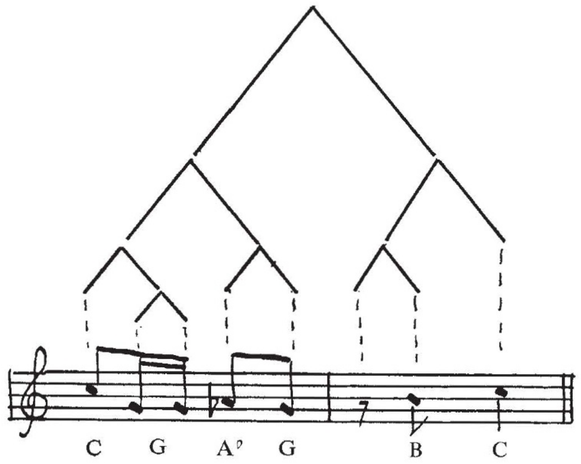
Centre for Research on Perception and Cognition, Laboratory of Experimental Psychology, University of Sussex, Brighton BN1 9QG, UK

A computer program has been written which will transcribe a live performance of a classical melody into the equivalent of standard musical notation. It is intended to embody, in computational form, a psychological theory of how Western musicians perceive the rhythmic and tonal relationships between the notes of such melodies.

A SEARCHING test of practical musicianship is the 'aural test' in which the subject is required to write down, in standard, musical notation, a melody which he has never heard before. His transcription is not to be construed as a detailed record of the actual performance, which will inevitably be more or less out of time and out of tune, but as an indication of the rhythmic and tonal relations between the individual notes. How the musical listener perceives these relationships is a matter of some interest to the cognitive psychologist. In this paper I outline a theory of the perception of classical Western melodies, and describe a computer program, based on the theory, which displays, as best it can, the rhythmic and tonal relationships between the notes of a melody as played by a human performer on an organ console.

The basic premise of the theory is that in perceiving a melody the listener builds a conceptual structure representing the rhythmic groupings of the notes and the musical intervals between them. It is this structure which he commits to memory, and which subsequently enables him to recognise the tune, and to reproduce it in sound or in writing if he happens to be a skilled musician. A second premise is that much can be learned about the structural relationships in any ordinary piece of music from a study of its orthographic representation. Take, for example, the musical cliché notated in Fig. 1.

Fig. 1



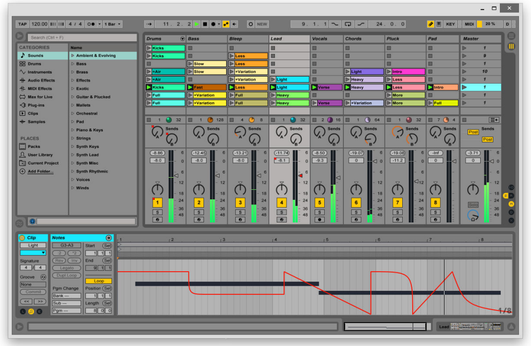
Automated Music Transcription today

Conversion of a recorded music performance into a music score

Audio recording

MIDI device
(score edition)

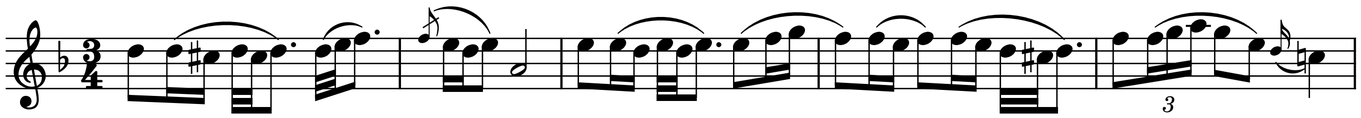
Algorithmic composition
DAW



audio Music Information Retrieval
- fundamental freq. estimation
- onset detection
- beat tracking …

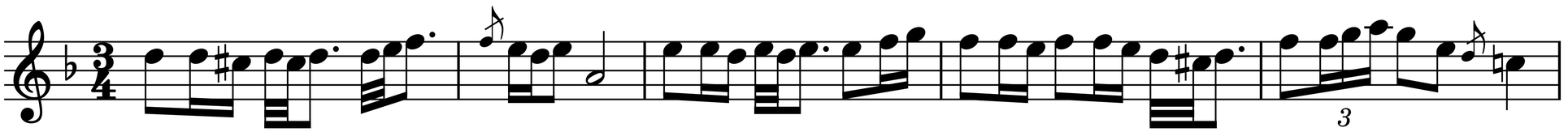
piano roll (MIDI file)
- unquantized onsets, durations
- quantized pitches

symbolic Music Information Retrieval
- rhythm quantization
- tempo tracking
- score engraving…

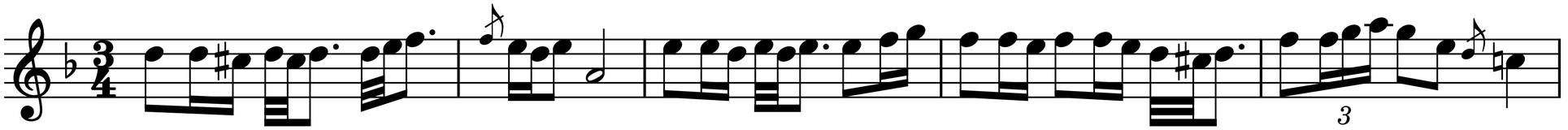


music score
(XML file)

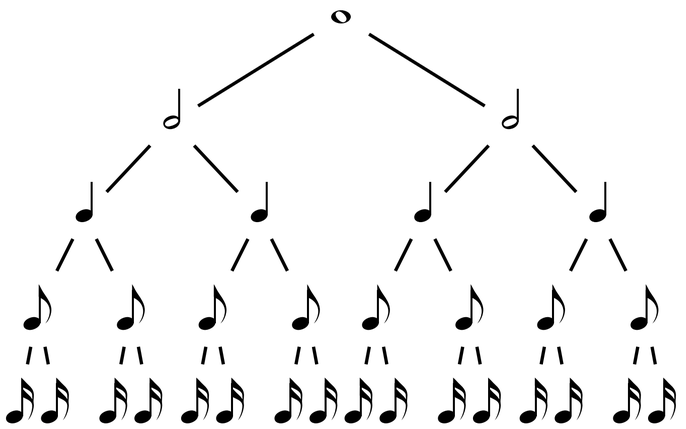
Polonaise in D minor from Notebook for Anna Magdalena Bach BWV Anh II 128



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

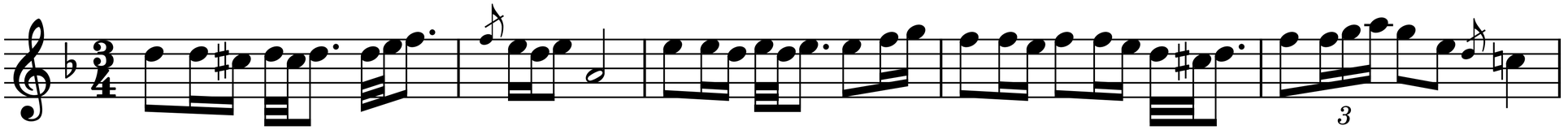


hierarchical
note
durations



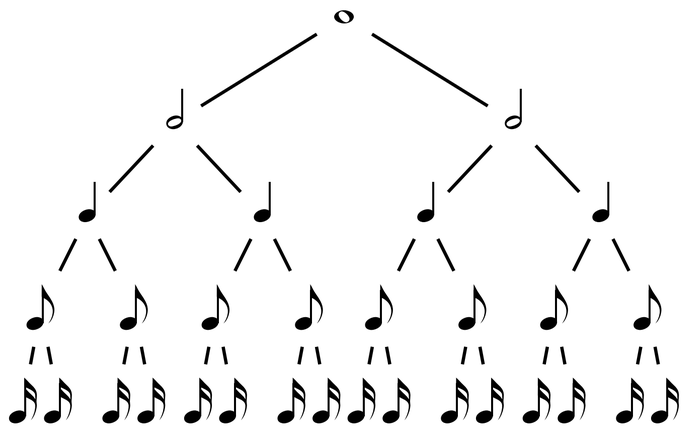
Polonaise in D minor from Notebook for Anna Magdalena Bach BWV Anh II 128

beamed



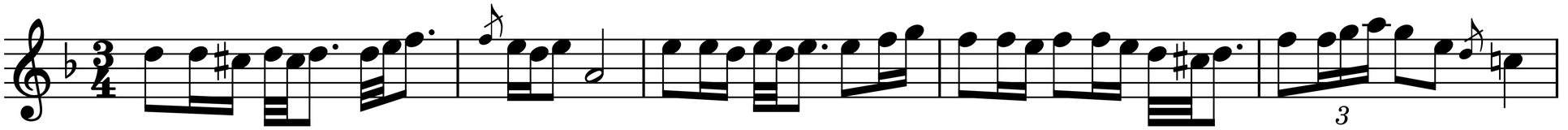
unbeamed

hierarchical note durations



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

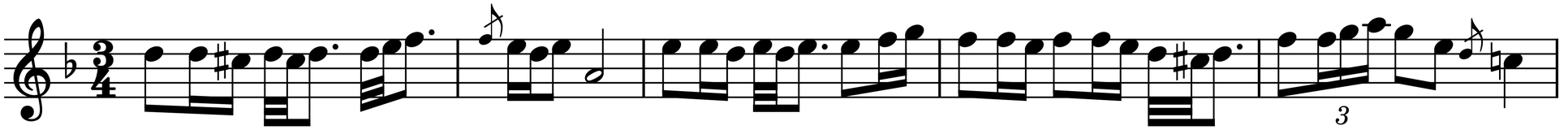
beamed



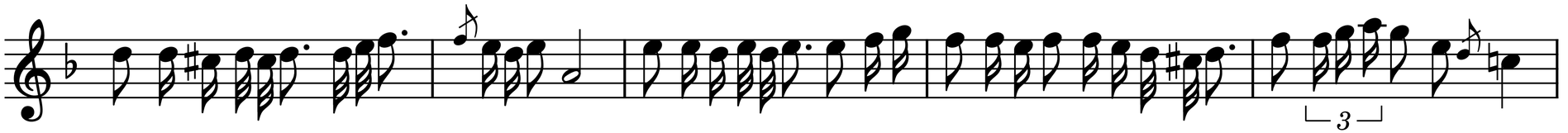
unbeamed

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

beamed



unbeamed

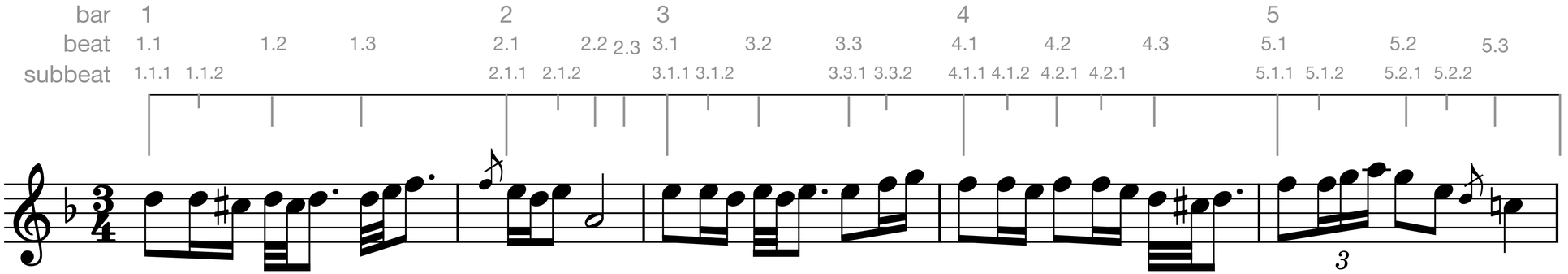


grouping notes with measure bars and beams

- eases readability (player reads in a real-time context)
- highlight the metric structure
  hierarchy of strong / weak beats

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

metric structure



beamed

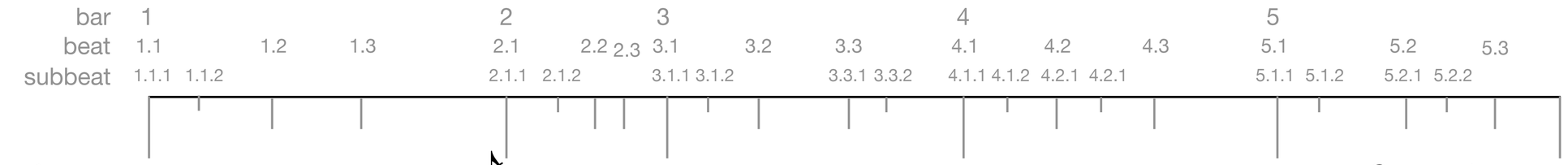
unbeamed

grouping notes with measure bars and beams

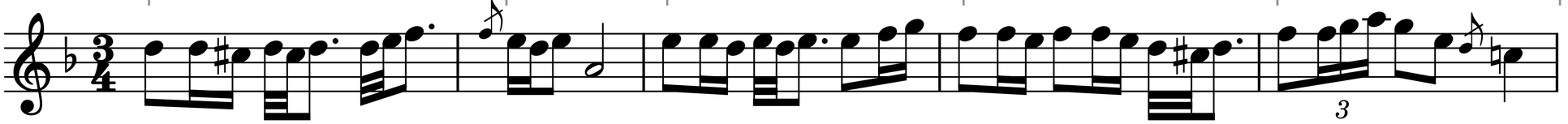
- eases readability (player reads in a real-time context)
- highlight the metric structure
  hierarchy of strong / weak beats

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

metric structure



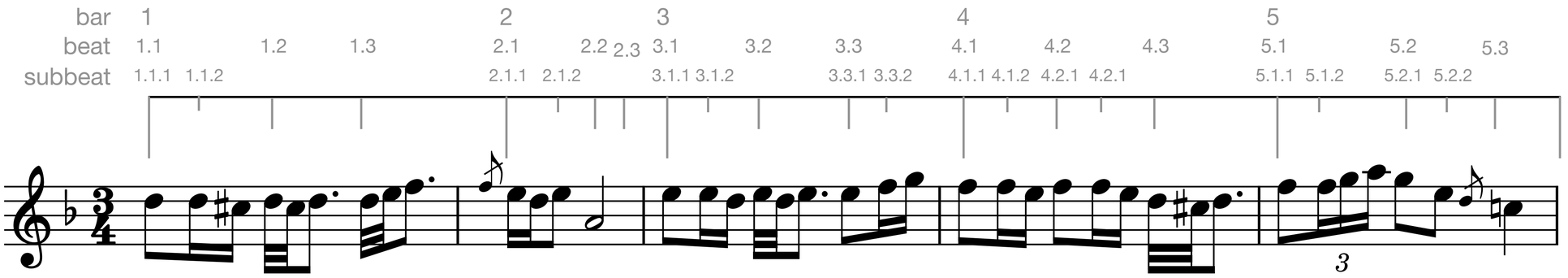
beamed



unbeamed

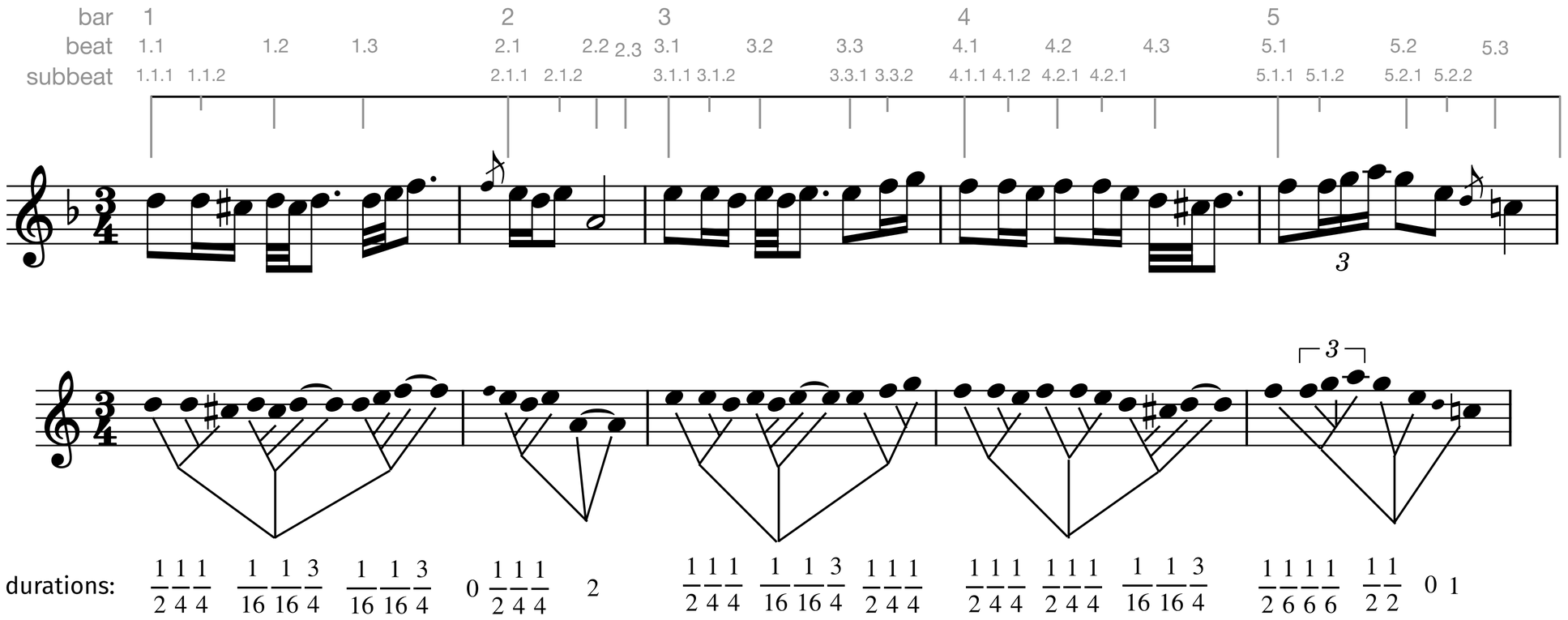
Polonaise in D minor from Notebook for Anna Magdalena Bach BWV Anh II 128

metric structure



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

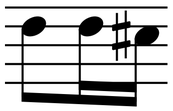
metric structure

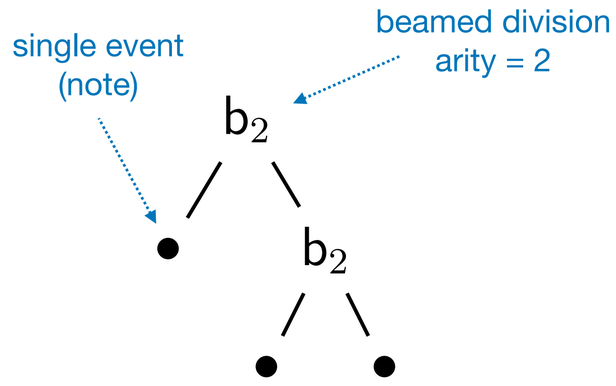


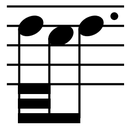
Tree representation of the proportional rhythmic notation

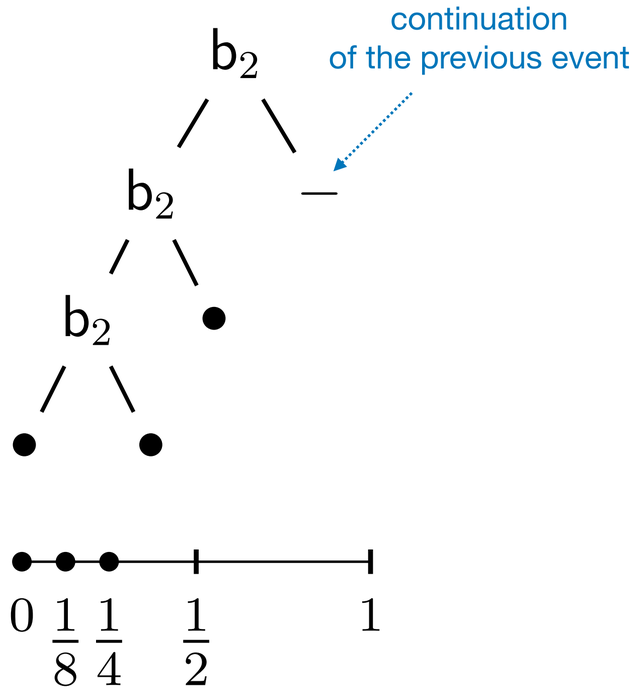
with hierarchical encoding of durations: “*the* (duration) *data is in the structure*”

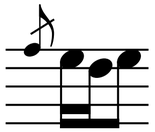
- the tree leaves contain the events
- the branching define durations, by partitioning of time intervals

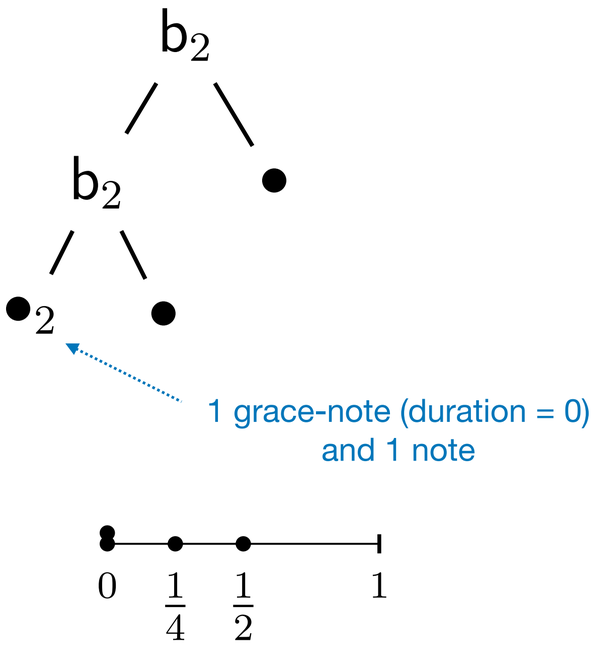




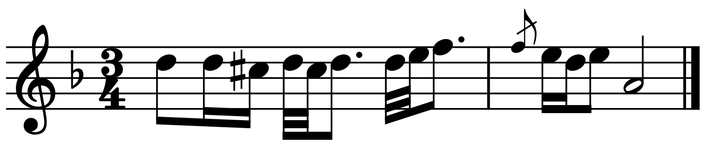


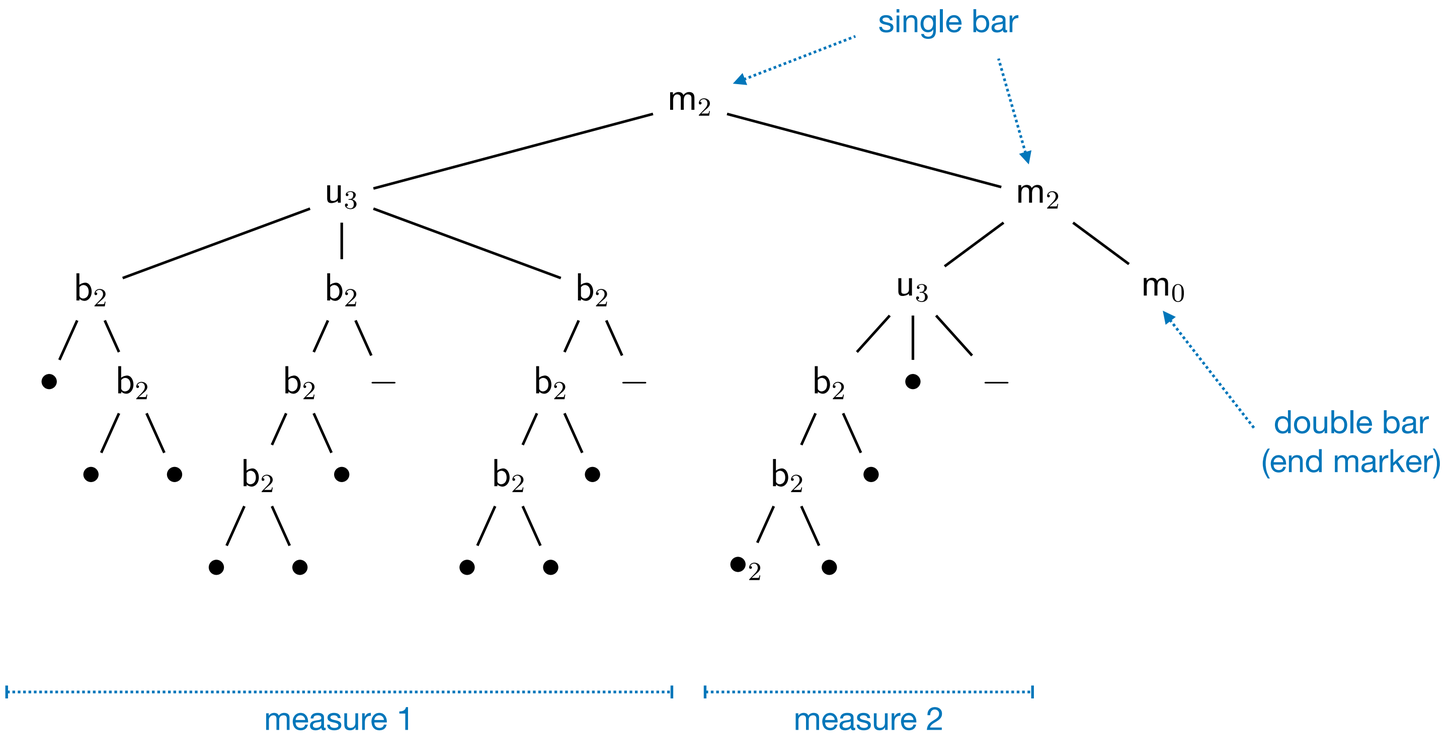






Tree-structured Representation of Music Notation





Regular Tree Language (of Music Notation)

defined by a Regular Tree Grammar:

- non-terminal symbols: $q, q_0, q_1, \ldots$
- terminal symbols (constants): • (1 note), $\bullet_2$ (1 grace-note + 1 note), — (continuation)
- production rules:

$$q \to \mathsf{m}_2(q_0, q) \mid \mathsf{m}_0$$

$$q_0 \to \mathsf{u}_3(q_1, q_1, q_1) \mid \bullet$$ measure

$$q_1 \to \mathsf{b}_2(q_2', q_2) \mid \bullet \mid \bullet_2 \mid -$$ beat = ♩

$$q_2' \to \mathsf{b}_2(q_3', q_3) \mid \bullet \mid \bullet_2 \mid -$$

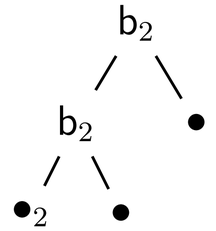
$$q_3' \to \bullet \mid \bullet_2 \mid -$$

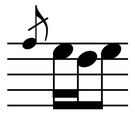
$$q_2 \to \mathsf{b}_2(q_3, q_3) \mid \bullet \mid -$$ sub-beat = 8th-note = ♪

$$q_3 \to \bullet \mid -$$ sub-sub-beat = 16nth note = 𝅘𝅥𝅯

derivations (lefmost)

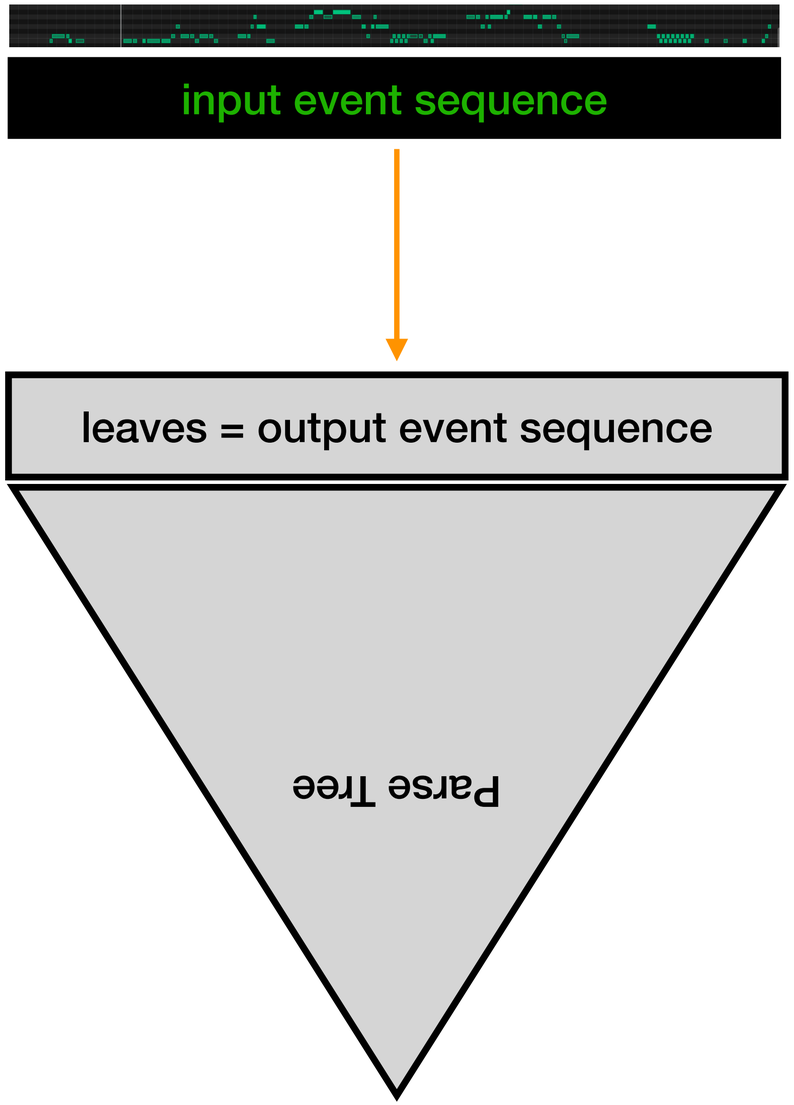
$$q_1 \to \mathsf{b}_2(q_2', q_2) \to \mathsf{b}_2(\mathsf{b}_2(q_3', q_3), q_2) \to \mathsf{b}_2(\mathsf{b}_2(\bullet_2, q_3), q_2) \to \mathsf{b}_2(\mathsf{b}_2(\bullet_2, \bullet), q_2) \to \mathsf{b}_2(\mathsf{b}_2(\bullet_2, \bullet), \bullet)$$





$$q \to \mathsf{m}_2(q_0, q) \to \mathsf{m}_2(\mathsf{u}_3(q_1, q_1, q_1), q) \to \mathsf{m}_2(\mathsf{u}_3(\mathsf{b}_2(q_2', q_2), q_1, q_1), q) \to \mathsf{m}_2(\mathsf{u}_3(\mathsf{b}_2(\bullet, q_2), q_1, q_1), q) \to \ldots$$

Transcription as Parsing



piano roll
= sequence of timestamped input events

tree-structured representation
of an output **music score**

conforming to a
prior language (expected notation)

Transcription as Parsing

input event sequence

piano roll
= sequence of timestamped input events

structuring a linear representation according to a language model

= **parsing**

leaves = output event sequence

Parse Tree

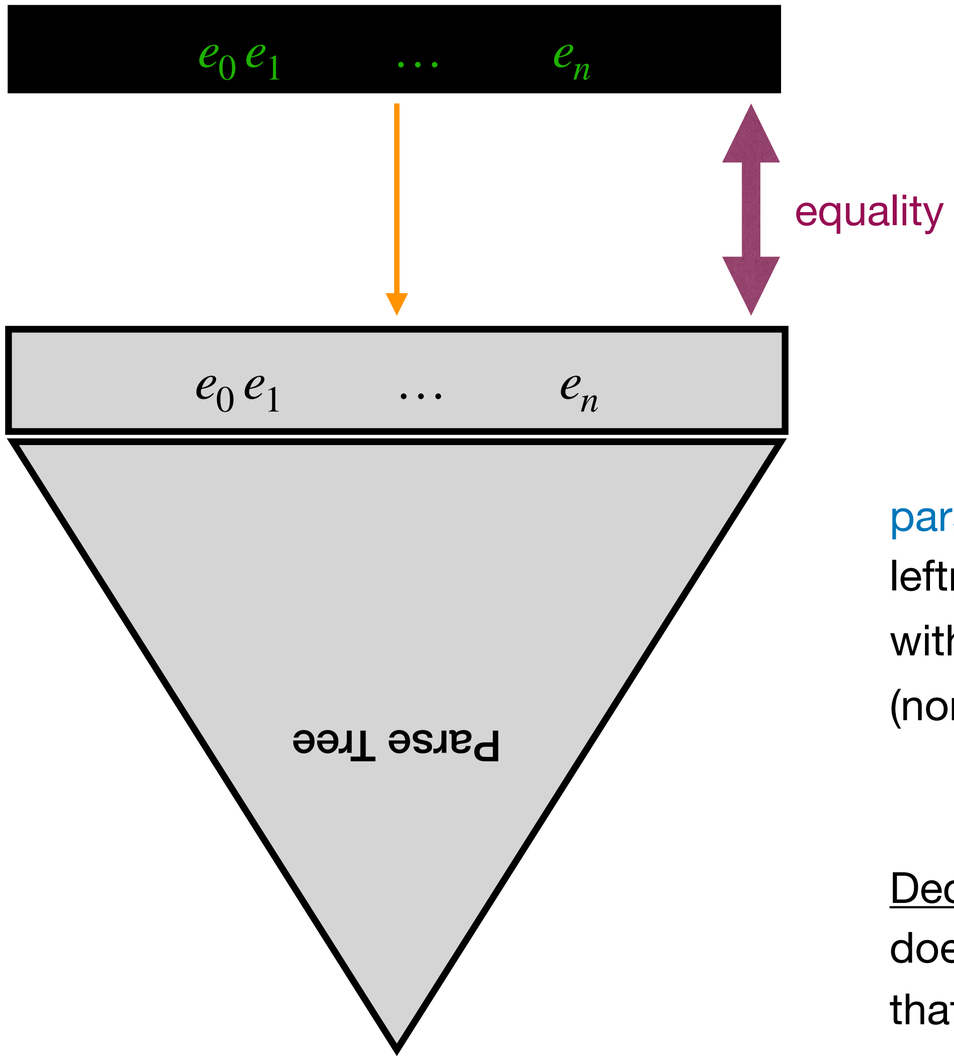
tree-structured representation
of an output **music score**

conforming to a
prior language (expected notation)

Conventional Parsing

terminal symbols: $e_0, \ldots$ in a finite alphabet

input sequence



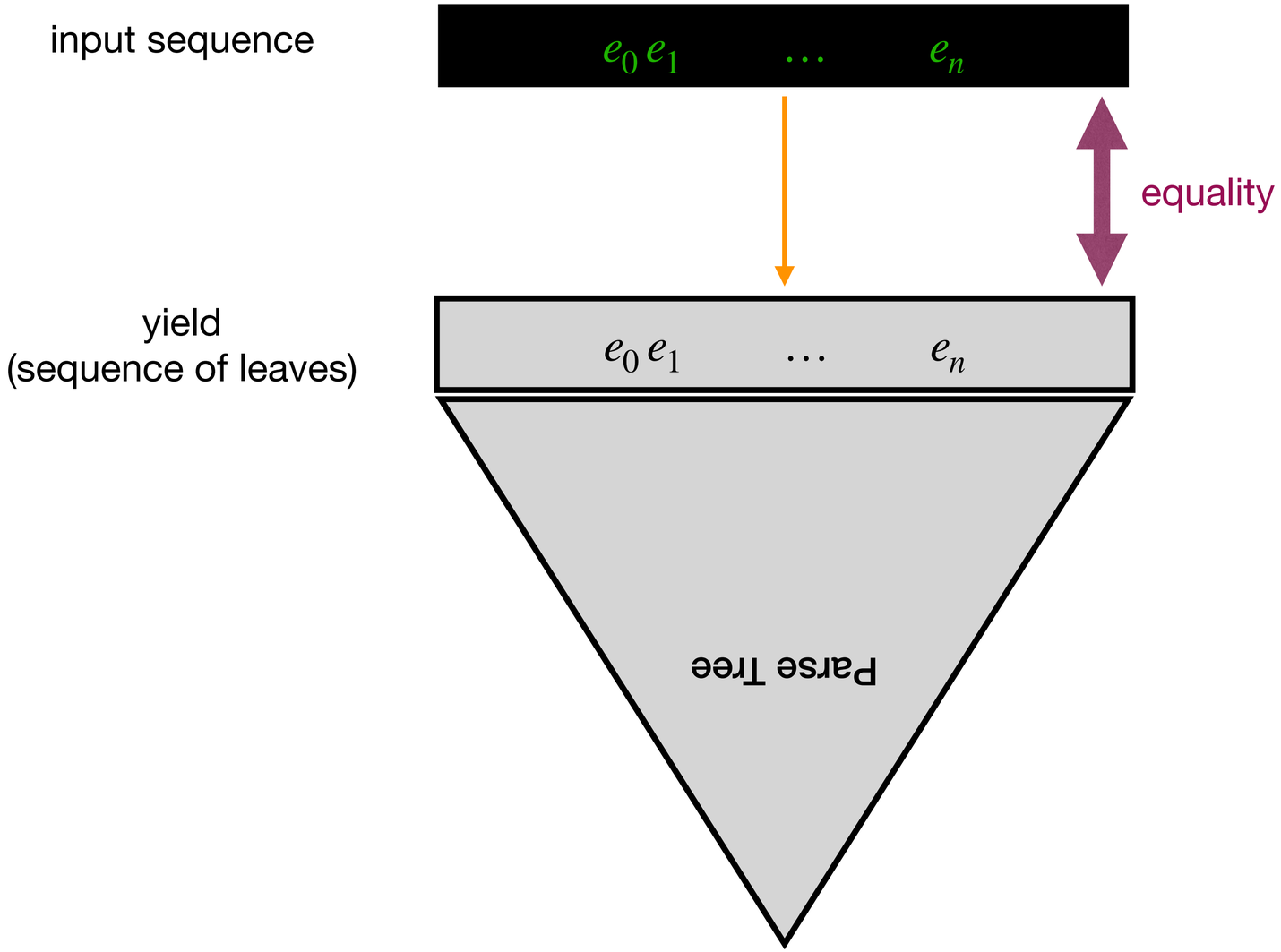
yield
(sequence of leaves)

parse-tree = representation of a
leftmost derivation of $e_0\, e_1 \ldots e_n$ by a prior CF-grammar $\mathcal{G}$
with production rules: $q_0 \to q_1\, q_2$ or $q_0 \to e$
(non-terminal symbols: $q_0, q_1, \ldots$)

Decision problem: (membership)
does there exists a parse tree (leftmost derivation) of $\mathcal{G}$
that yields $e_0\, e_1 \ldots e_n$?

Weighted Parsing

Returning a parse tree of $\mathscr{G}$ that yields $e_0\, e_1 \ldots e_n$



With an ambiguous prior CF-grammar $\mathscr{G}$ there might exists several parse trees (exponentially many).

in order to choose one (or some) parse trees, rank them according to their weight values, computed by Weighted Tree Grammar

Weighted Tree Language (*tree series*)

Weighted Regular Tree Grammar $\mathcal{G}$:

- non-terminal symbols: $q, q_0, q_1, \ldots$
- terminal symbols (constants): $\bullet$ (1 note), $\bullet_2$ (1 grace-note + 1 note), $-$ (continuation)
- every production rule is assigned a weight value (*e.g.* cost to read):

$$q \xrightarrow{0} \mathsf{m}_2(q_0, q) \qquad q \xrightarrow{0} \mathsf{m}_0$$

$$q_0 \xrightarrow{0.1} \mathsf{u}_3(q_1, q_1, q_1) \qquad q_0 \xrightarrow{1} \bullet$$

$$q_1 \xrightarrow{0.1} \mathsf{b}_2(q_2', q_2) \qquad q_1 \xrightarrow{1} \bullet \qquad q_1 \xrightarrow{1.9} \bullet_2 \qquad q_1 \xrightarrow{1} -$$

$$q_2' \xrightarrow{0.1} \mathsf{b}_2(q_3', q_3) \qquad q_2' \xrightarrow{1} \bullet \qquad q_2' \xrightarrow{2.25} \bullet_2 \qquad q_2' \xrightarrow{1} -$$

$$q_2 \xrightarrow{0.1} \mathsf{b}_2(q_3, q_3) \qquad q_2 \xrightarrow{1} \bullet \qquad\qquad q_2 \xrightarrow{1} -$$

$$q_3' \xrightarrow{1} \bullet \qquad q_3' \xrightarrow{3.25} \bullet_2 \qquad q_3' \xrightarrow{1} - \qquad q_3 \xrightarrow{1} \bullet \qquad q_3 \xrightarrow{1} -$$

measure

beat = ♩

sub-beat = 8th-note = ♪

sub-sub-beat = 16th note = 𝅘𝅥𝅯

derivation (lefmost):

$$d : q_1 \xrightarrow{0.1} \mathsf{b}_2(q_2', q_2) \xrightarrow{0.1} \mathsf{b}_2(\mathsf{b}_2(q_3', q_3), q_2) \xrightarrow{3.25} \mathsf{b}_2(\mathsf{b}_2(\bullet_2, q_3), q_2) \xrightarrow{1} \mathsf{b}_2(\mathsf{b}_2(\bullet_2, \bullet), q_2) \xrightarrow{1} \mathsf{b}_2(\mathsf{b}_2(\bullet_2, \bullet), \bullet)$$

cost of derivation: $\mathsf{weight}(d) = 0.1 + 0.1 + 3.25 + 1 + 1$

learning weight values from corpus statistics - Francesco Foscarin

Weight Values

In general, the weight values are taken in a commutative Semiring $\langle \mathbb{S}, \oplus, \mathbb{O}, \otimes, \mathbb{I} \rangle$

- $\oplus$ and $\otimes$ are associative and commutative, with neutral elements $\mathbb{O}$ and $\mathbb{I}$
- $\otimes$ distributes over $\oplus$: $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$
- $\mathbb{O}$ is absorbing for $\otimes$: $\mathbb{O} \otimes x = \mathbb{O}$

	domain	$\oplus$	$\otimes$	$\mathbb{O}$	$\mathbb{I}$
Boolean	$\{ \bot, \top \}$	$\vee$	$\wedge$	$\bot$	$\top$
Viterbi	$[0,1] \subset \mathbb{R}$	max	$\times$	0	1
Tropical min-plus	$\mathbb{R}_+ \cup \{+\infty\}$	min	$+$	$+\infty$	0

Moreover, $\oplus$ is assumed to extend to infinite sums.

$\otimes$ is for composition of rule's weights in derivations and $\oplus$ is for optimal choice:
For a Weighted Regular Tree Grammar $\mathcal{G}$

$$\text{weight}_{\mathcal{G}}(d : q \xrightarrow{w_1} \dots \xrightarrow{w_n} t) = \bigotimes_{i=1}^{n} w_i \quad \text{and} \quad \text{weight}_{\mathcal{G}}(q, t) = \bigoplus_{d : q \xrightarrow{+} t} \text{weight}_{\mathcal{G}}(d)$$

or recursively:

$$\text{weight}_{\mathcal{G}}(q, a(t_1, \dots, t_n)) = \bigoplus_{q \xrightarrow{w} a(q_1, \dots, q_n) \in \mathcal{G}} \left(w \otimes \bigotimes_{i=1}^{n} \text{weight}_{\mathcal{G}}(q_i, t_i) \right)$$

	domain	$\oplus$	$\otimes$	$\mathbb{0}$	$\mathbb{1}$
Boolean	$\{ \bot , \top \}$	$\vee$	$\wedge$	$\bot$	$\top$
Viterbi	$[0,1] \subset \mathbb{R}$	max	$\times$	0	1
Tropical min-plus	$\mathbb{R}_+ \cup \{+\infty\}$	min	$+$	$+\infty$	0

$\mathbb{S}$ is assumed :

- idempotent $x \oplus x = x$
  that induces a partial ordering: $x \leq_\oplus y$ iff $x \oplus y = x$
- total : $\forall x, y \in \mathbb{S}$, either $x \oplus y = x$ or $x \oplus y = y$ *i.e.* $\leq_\oplus$ is total
- bounded : $\mathbb{1} \oplus x = \mathbb{1}$, or equivalently: $\forall x, y \in \mathbb{S}, x \leq_\oplus x \otimes y$
  i.e. combining elements with $\otimes$ always increases their weight,
  see the *non-negative weights* condition for Dijkstra's shortest path algorithm

k-best parsing : enumeration of the k best weighted trees *wrt* $\leq_\oplus$ for $\mathcal{G}$ and a non-terminal q, in PTIME.

similar to best path search in hyper-graphs (Dynamic Programming)

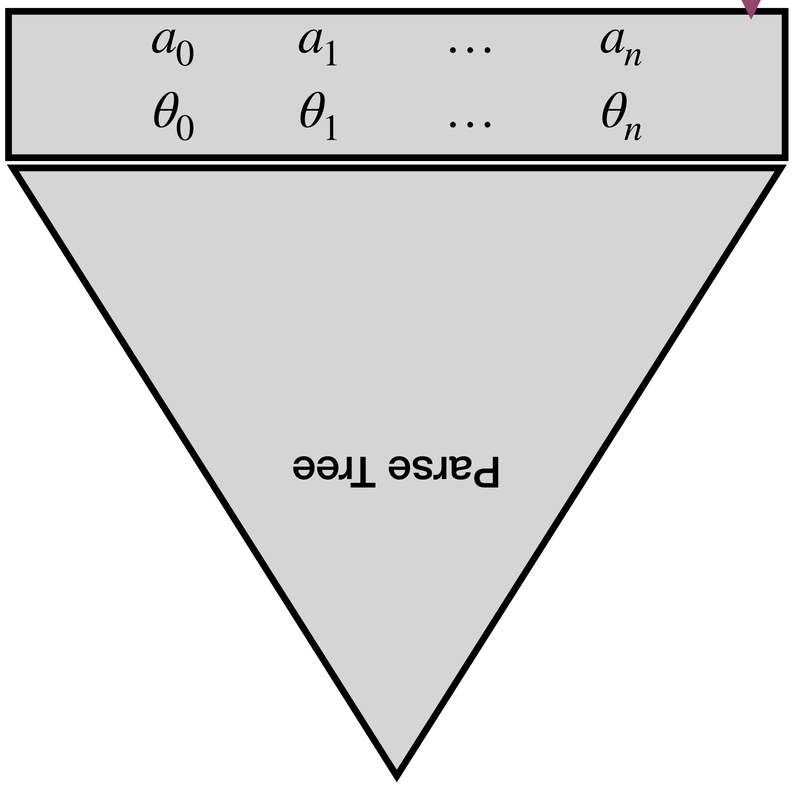
- Viterbi algorithm in acyclic case
- Knuth generalization of the Dijkstra algorithm in the general case

Quantitative Parsing

in the context of music transcription, the symbols in the sequence in input are timestamped

input sequence

$\neq$



yield
= sequence of leaves
decorated with dates θ
(computed with the durations
encoded in the tree structure)

ranking of solutions with:

a measure of input / output fitness
= cost of IO alignement

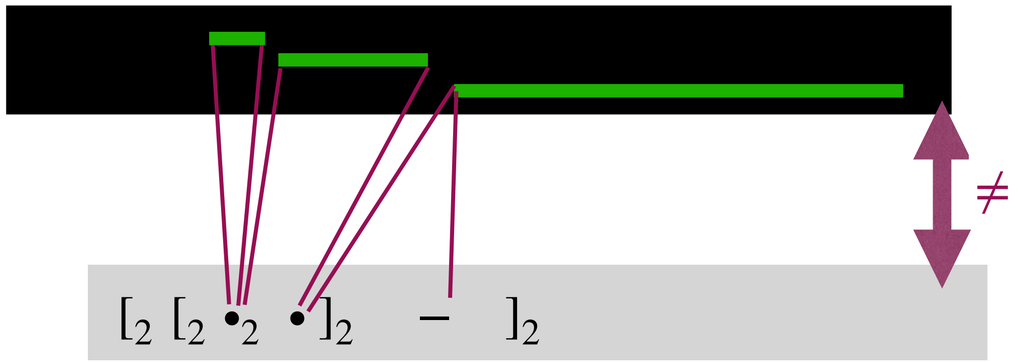
$\otimes$

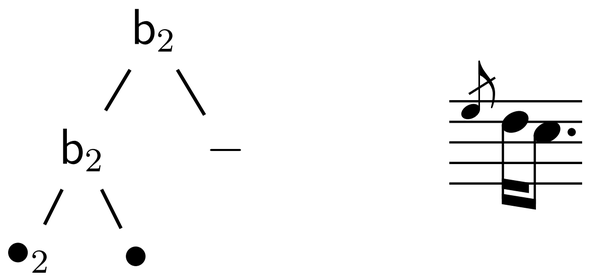
a measure of cost-to-read
(weight value computed by Weighted Tree Grammar)

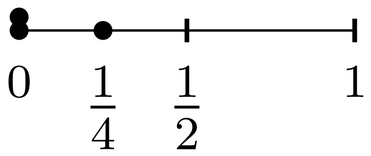
measure of input/output fitness

input sequence

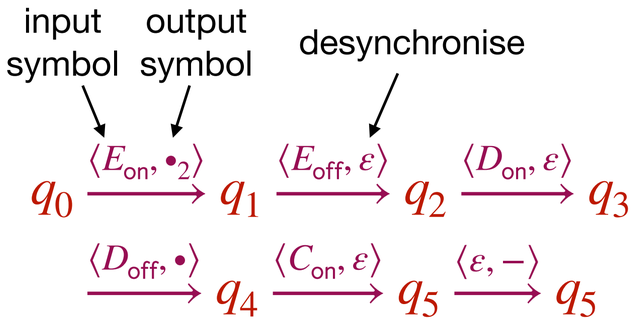
linearisation
of the output tree



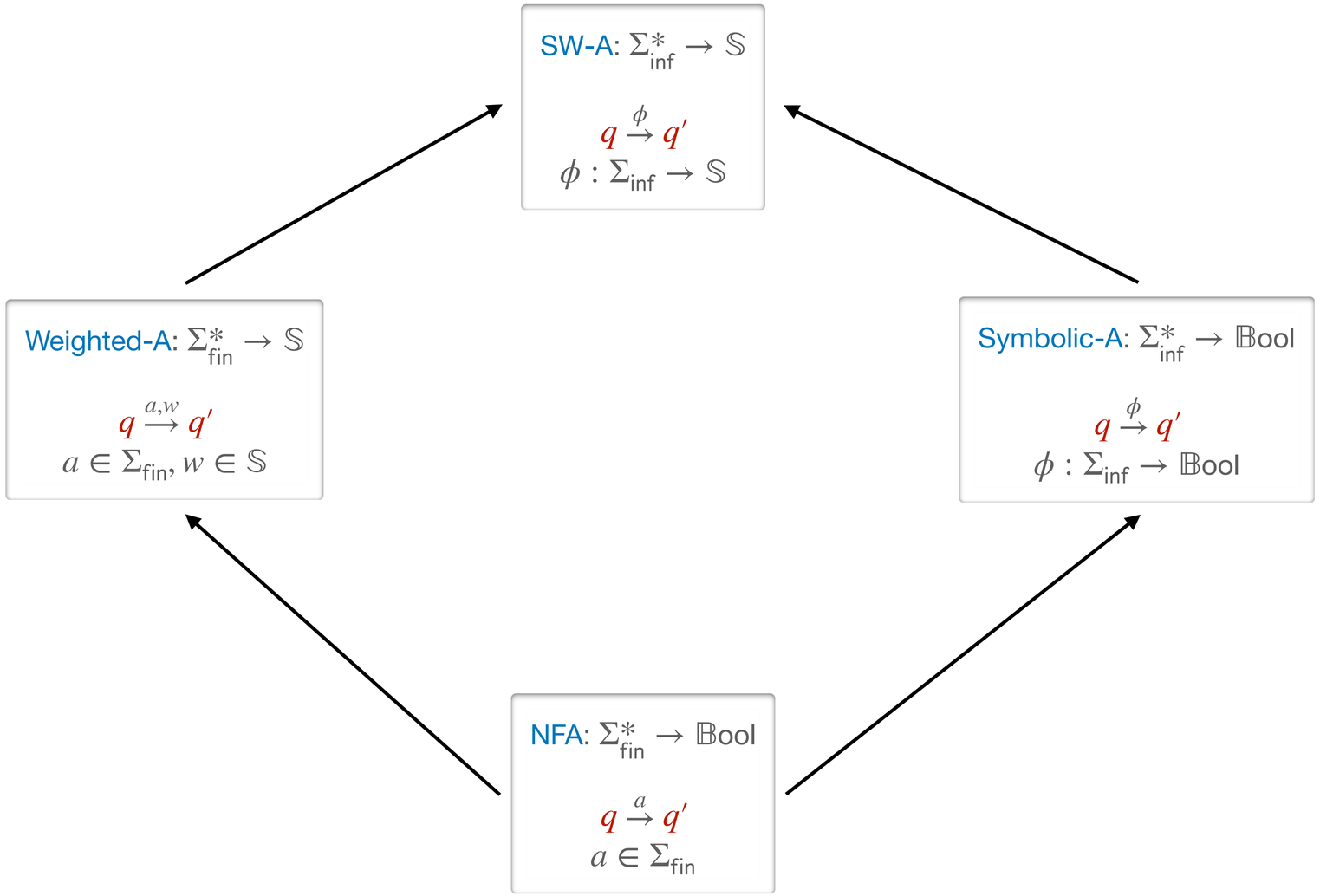




cost of IO alignement
computed by a
word-to-word Transducer
(stateful definition of an edit-distance)



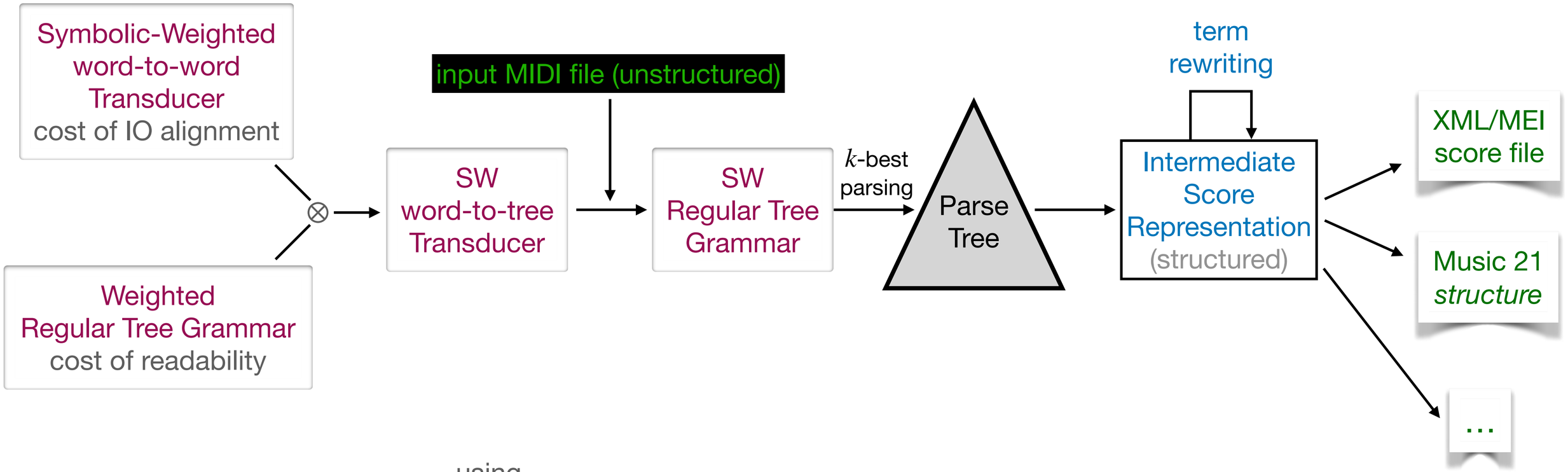
in the context of music transcription, the symbols are timestamped → infinite alphabet Σ_{inf}



Droste, M., Kuich, W., Vogler
Handbook of WA, 2009

Veanes et al.
CAV 2017, CACM 2021

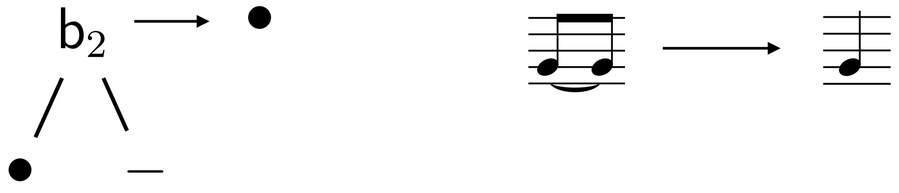
Automated Music Transcription: qparse framework

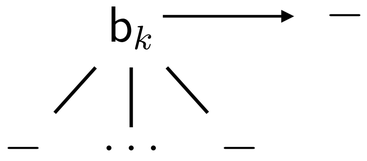


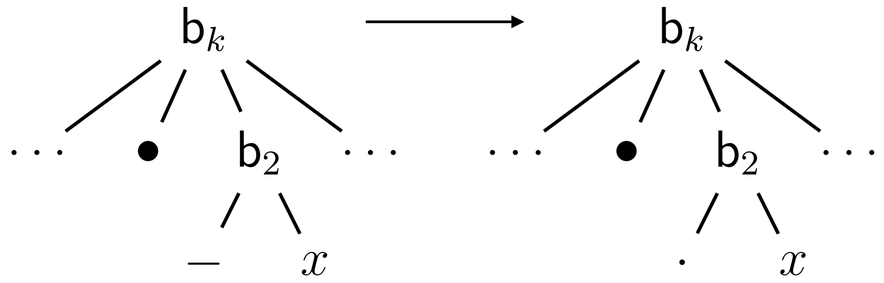
using
SW Visibly Pushdown Automata
as intermediate model

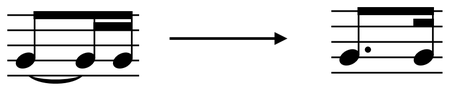
Term Rewriting Rules

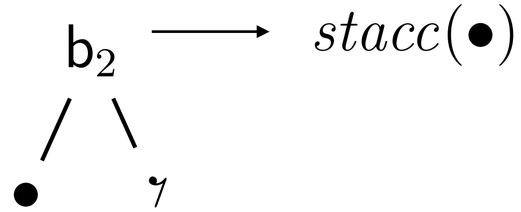
for the transformation of the intermediate score representation

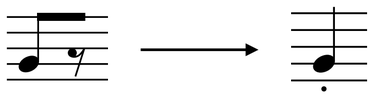












questions: rewrite strategies (*e.g.* IO or OI), conflicts…

implementation of

- the above transcription by parsing framework
- + other subtasks: pitch-spelling, key estimation, beat tracking…

qparse: 75 Kloc C++

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

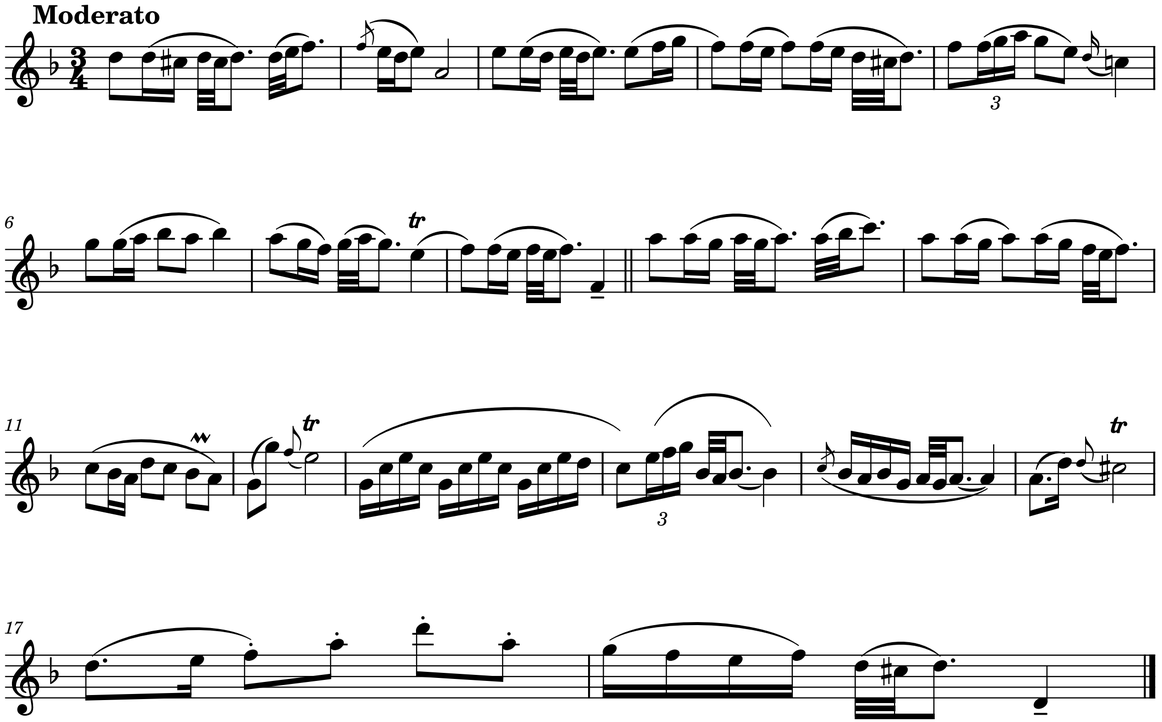
- command lines tools:
  monoparse, drumparse, grammar-learning, engraving
- Python binding - Lydia Rodrigez-de la Nava
  automatic evaluation
- online port, real-time - Leyla Villaroel

Monophonic transcription

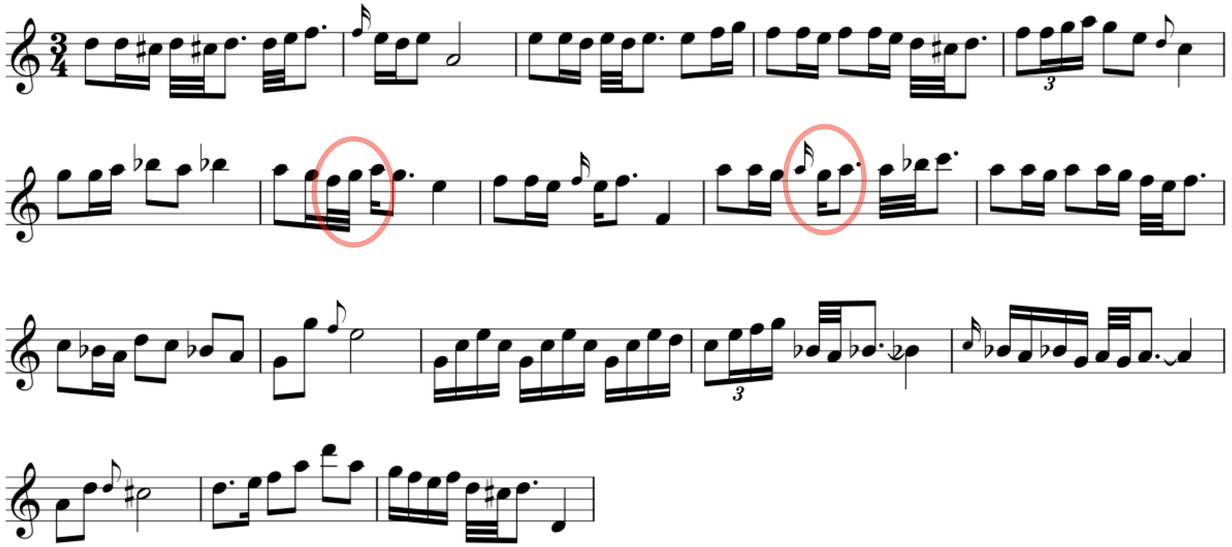
monophonic : one note at a time
Good results for complex cases (tuplets, ornaments, silences, small durations…)

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

original score



transcription of MIDI recording by qparse

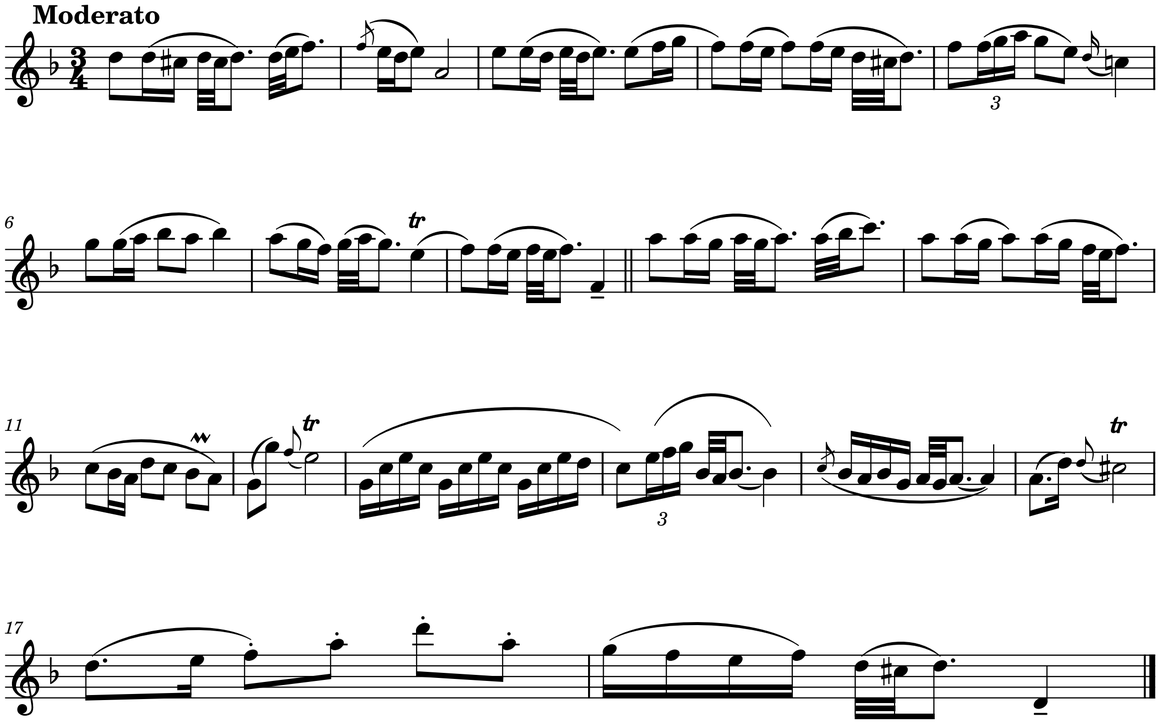


Monophonic transcription

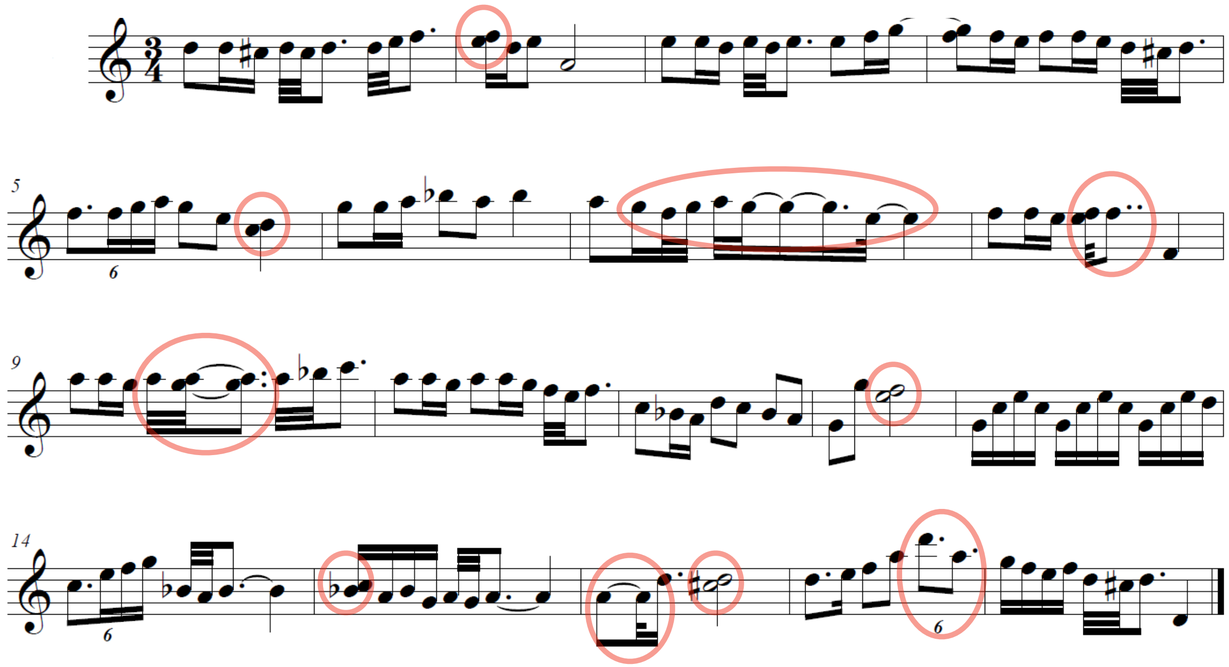
monophonic : one note at a time
Good results for complex cases (tuplets, ornaments, silences, small durations…)

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

original score



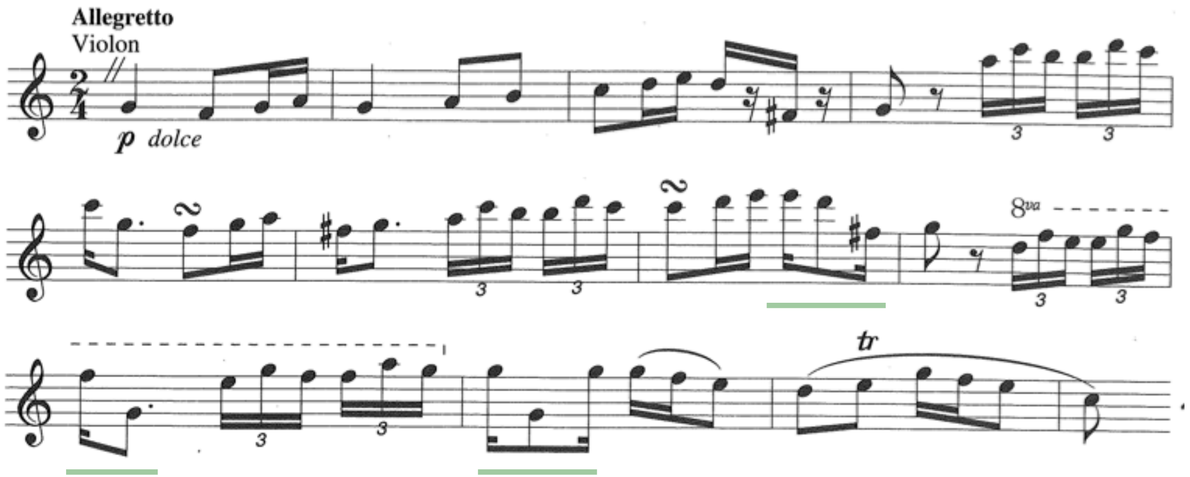
transcription of MIDI recording by Finale



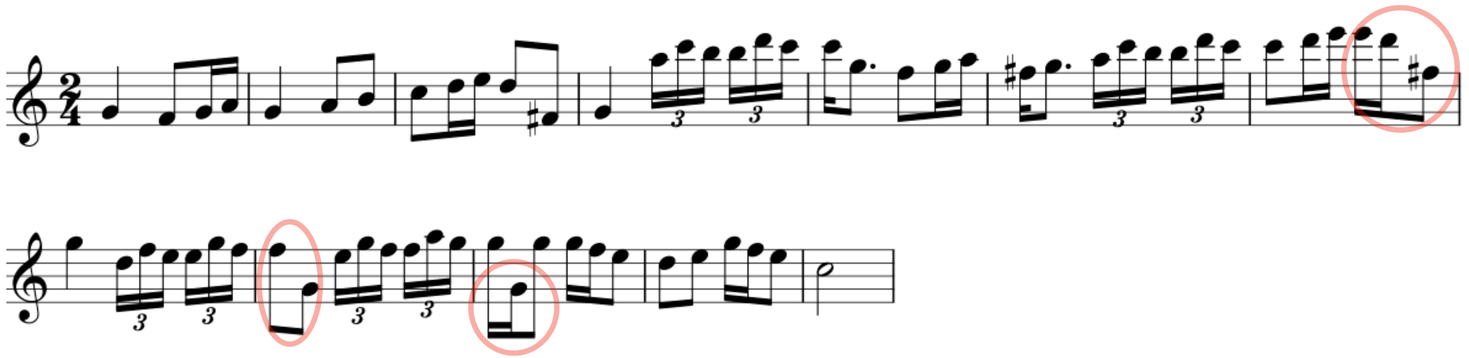
Monophonic transcription

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

original score



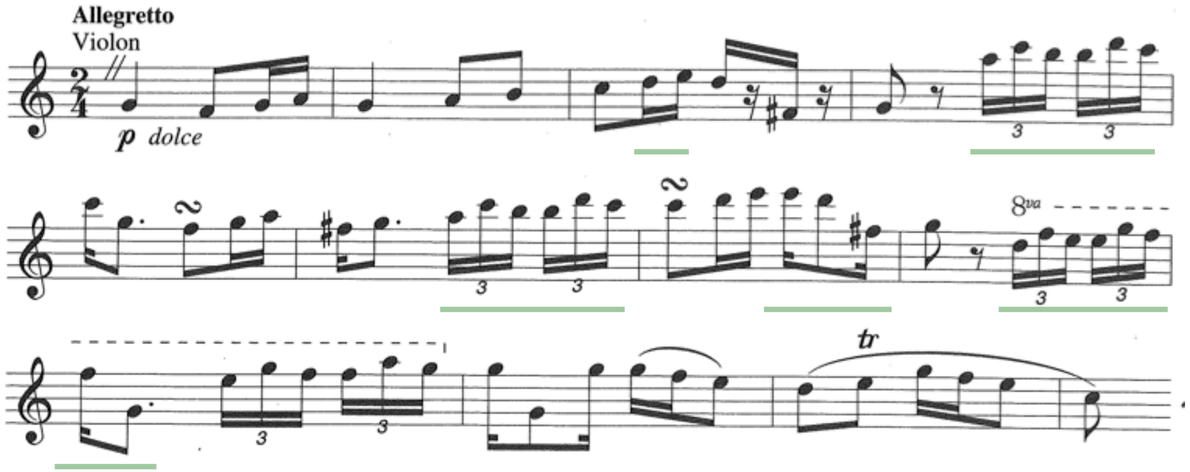
transcription of MIDI recording by qparse



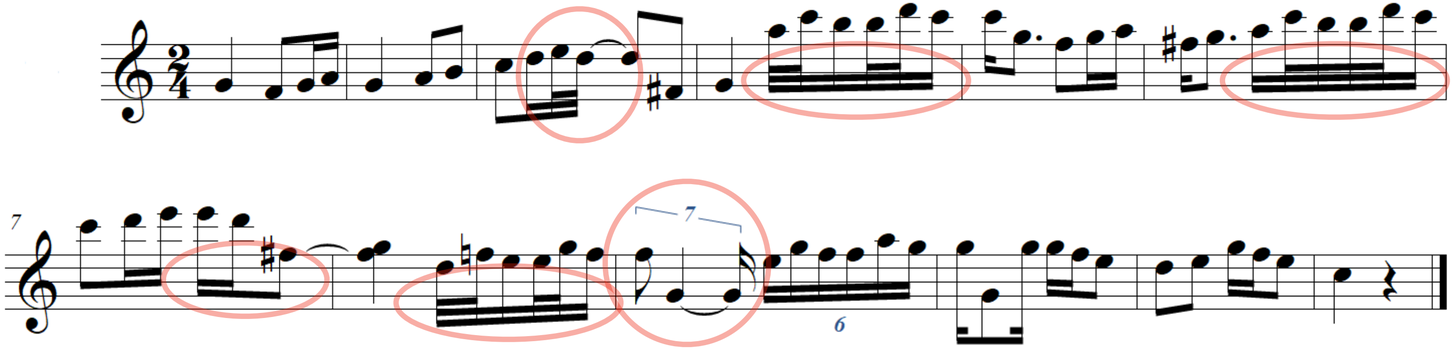
Monophonic transcription

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

original score



transcription
of MIDI recording
by Finale

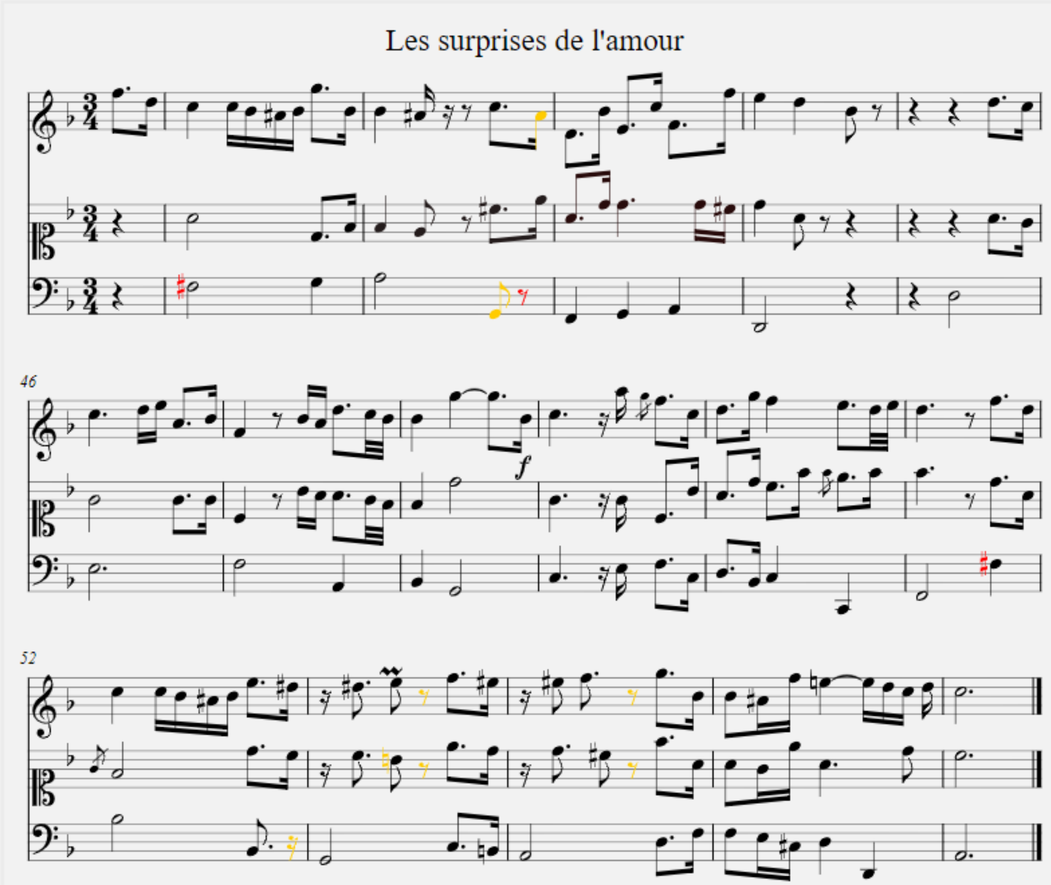


options:
- mixed rhythms,
- tuplets
- smallest note = 32nd

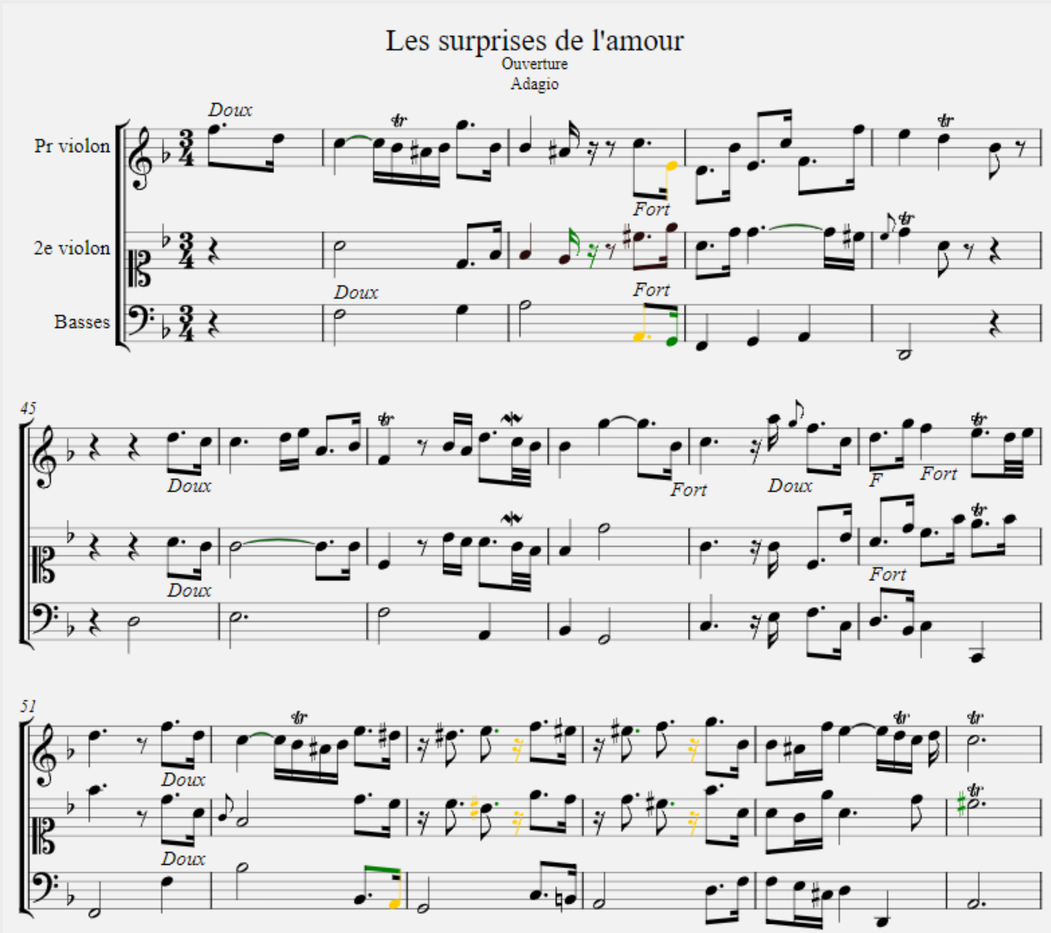
The time signature
and the tempo are given.

Francesco Foscarin

OMRized version



Manual correction (ground truth)

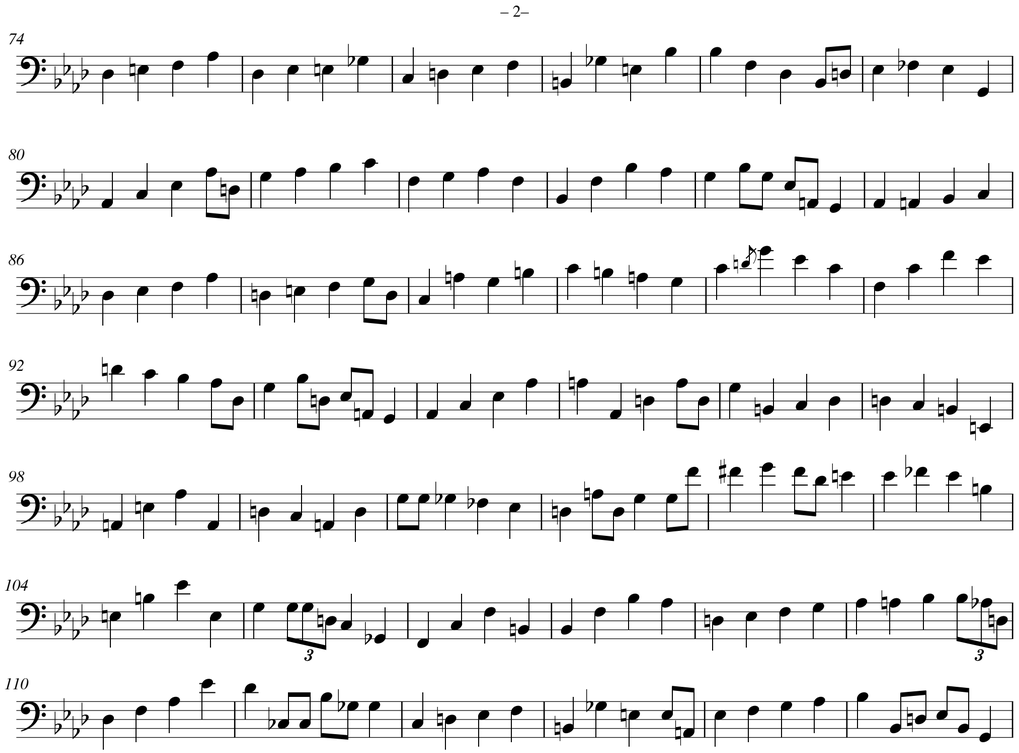


Lamarque-Goudard dataset (w. Francesco Foscarin, Teysir Baoueb)

- 283 monophonic extracts
  of classical repertoire
  inspired by a rhythm learning method
- ~ 20 measures per extract
- progressive difficulty
  cover a very large spectrum of rhythmic features
- score files (XML) and MIDI performances
  for evaluation and calibration of transcription tools

FiloBass by John-Xavier Riley (QMUL, C4DM)
project "*Dig That Lick*"

- jazz bass lines, acc. of saxophone
- 48 tracks,
  24 recorded hours of melodies and improvisations
- qparse as backend of an audio-MIDI transcription



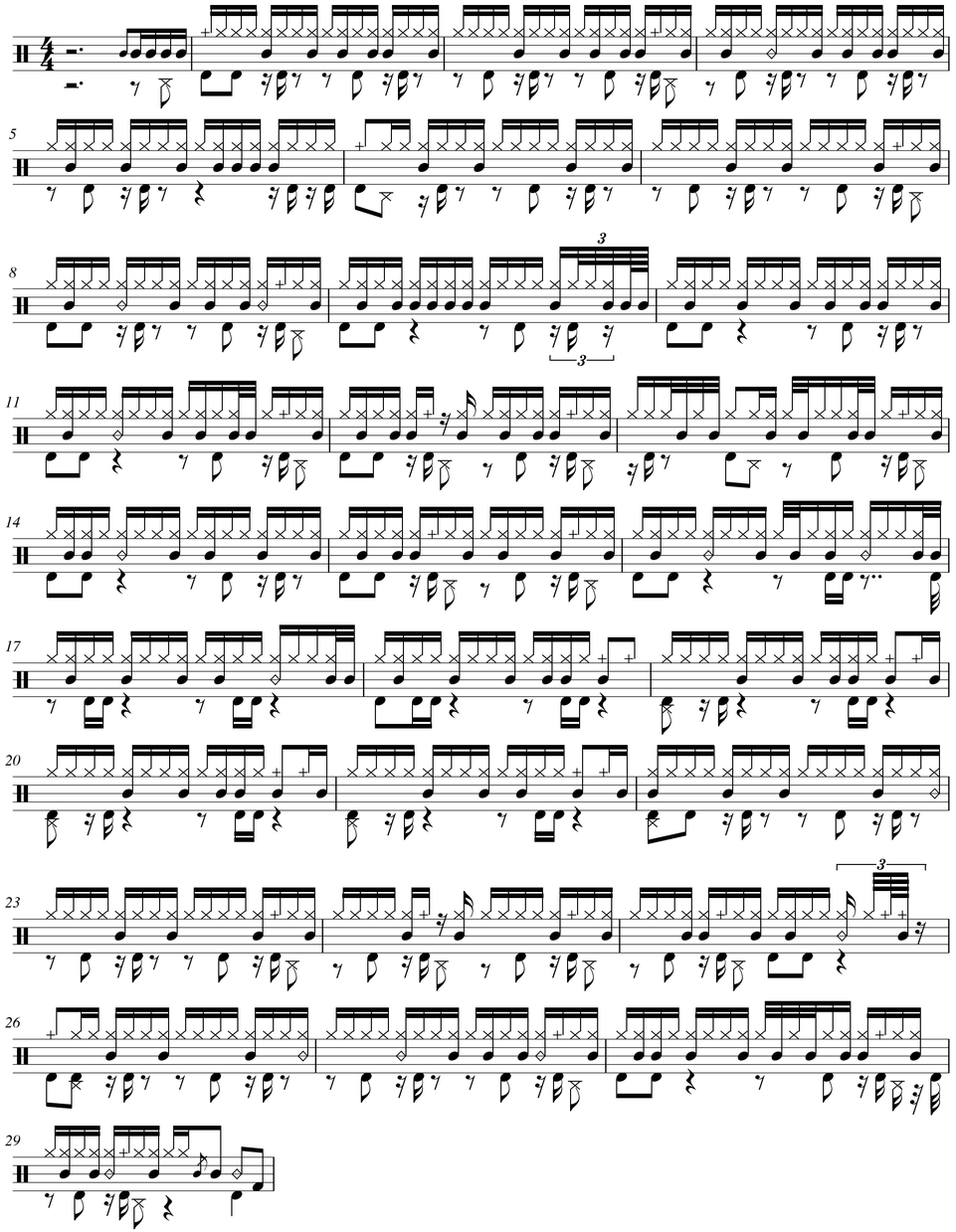
Groove MIDI Dataset

- by Google Magenta
- 13.6 hours, 1150 MIDI files, ~ 22000 measures recorded by professional drummers on a electronic drum kit
- audio (wav) files synthesized from (and aligned to) MIDI files for evaluation of audio-to-MIDI drum transcription
- no score files!

scoring the GMD with qparse

Martin Digard (INALCO)

- all score files (XML) produced from the MIDI files with the same generic tree grammar (4/4 measure)
- polyphonic case-study, simpler than piano
- specific drumming constraints (hands ≤ 2, feet ≤ 2)
- processing errors from MIDI sensors



Piano transcription

- Dataset **ASAP** - Francesco Foscarin, Andrew Mc Leod
  MIDI and audio recording from Yamaha piano competition
  + XML scores
  + alignments
  + beat tracking annotations

- voice separation - Lydia Rodrigez-de la Nava, evaluation Augustin Bouquillard
  and for piano guitar transcription.
  integration in transcription:
  - before parsing, or
  - after parsing (on intermediate model), or
  - joint with parsing.