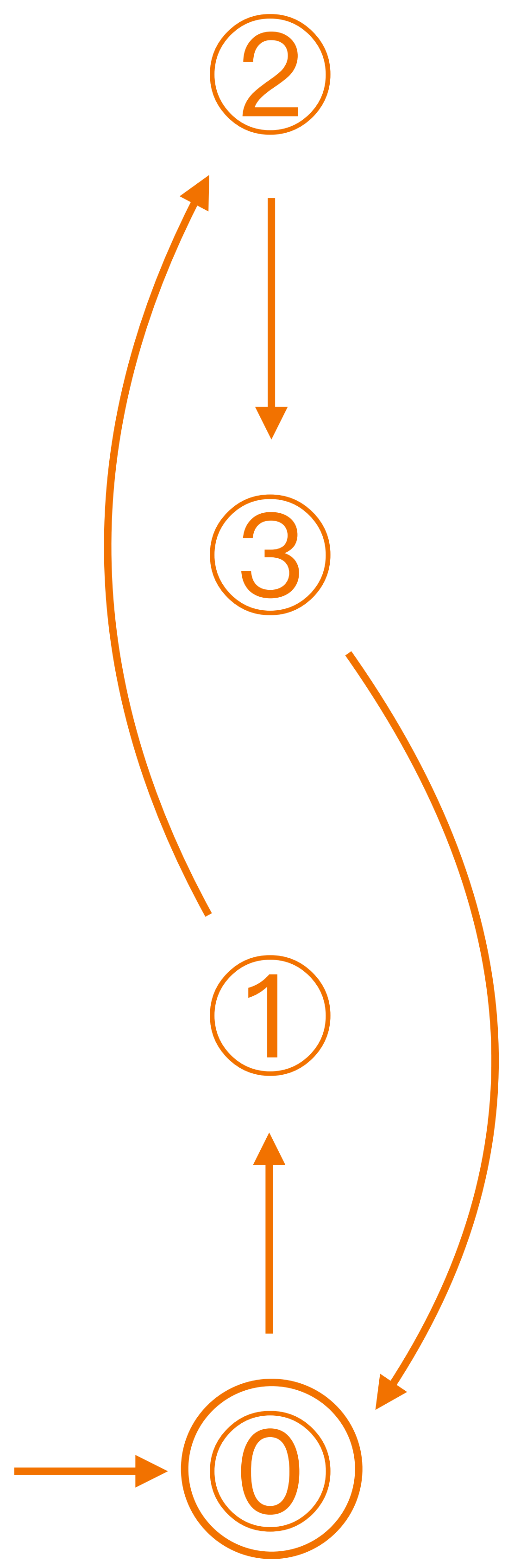
Symbolic Weighted Language Models, Quantitative Parsing and Automated Music Transcription

Florent Jacquemard and Lydia Rodriguez-de la Nava

Inria informatiques mathématiques
le cnam

CIAA 2022 - Rouen

Outline



Models: Symbolic and Weighted Visibly Pushdown Automata ad Transducers

Results: composition, partial application, 1-best search

Application: Quantitative Parsing

Application^2: Automated Music Transcription

Automated Music Transcription

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

articles

Perception of melodies

H. C. Longuet-Higgins

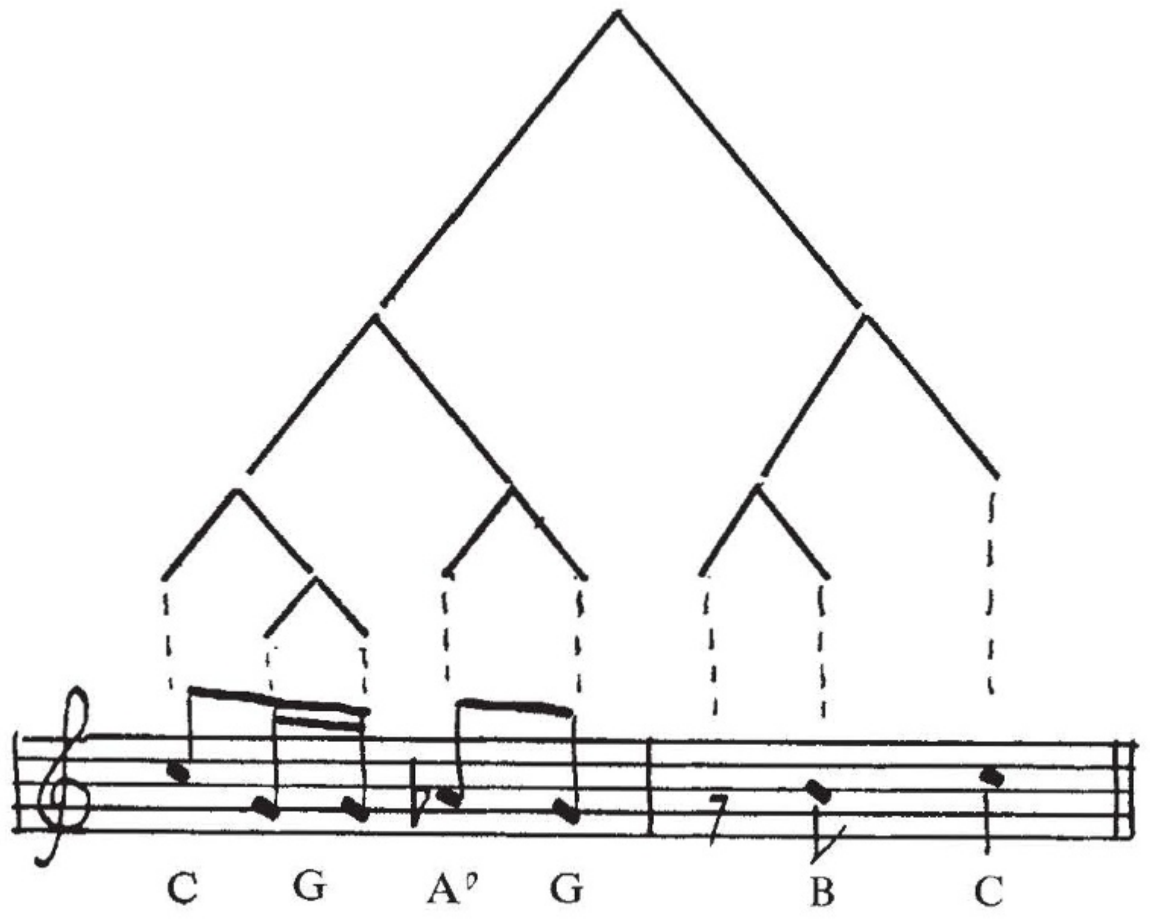
Centre for Research on Perception and Cognition, Laboratory of Experimental Psychology, University of Sussex, Brighton BN1 9QG, UK

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

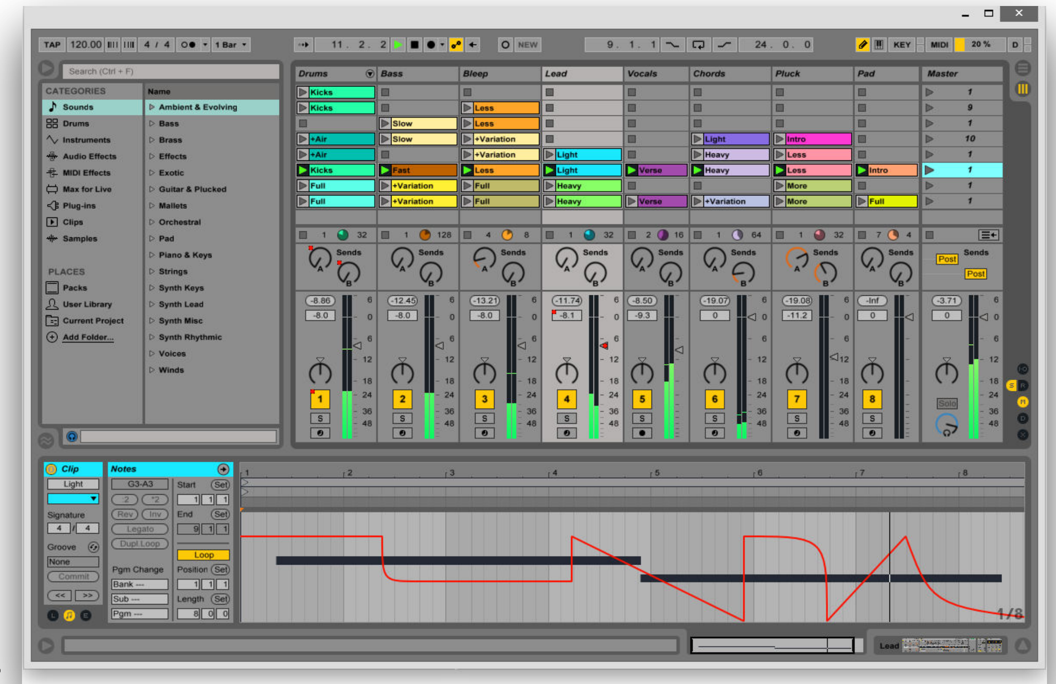
Conversion of a recorded music performance into a music score

source(s)

Audio recording

MIDI device
(score edition)

Algorithmic composition
DAW

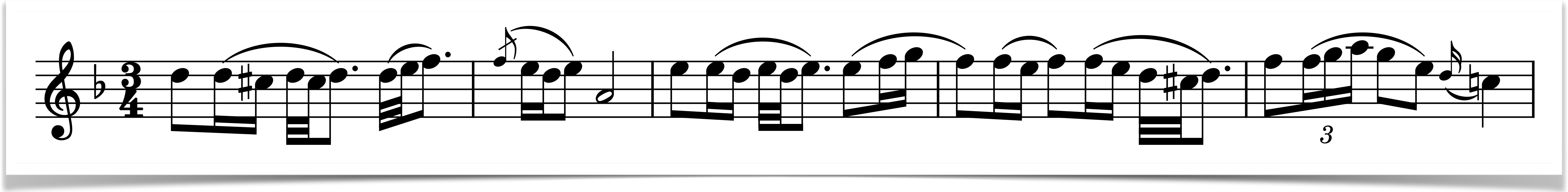


audio Music Information Retrieval
- fundamental freq. estimation
- onset detection
- beat tracking …

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

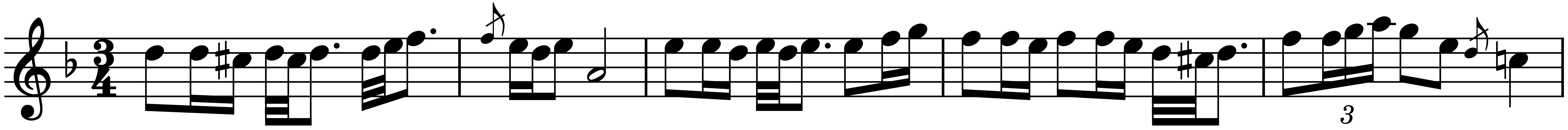
symbolic Music Information Retrieval
- rhythm quantization
- tempo tracking
- score engraving…

target
music score
(XML file)

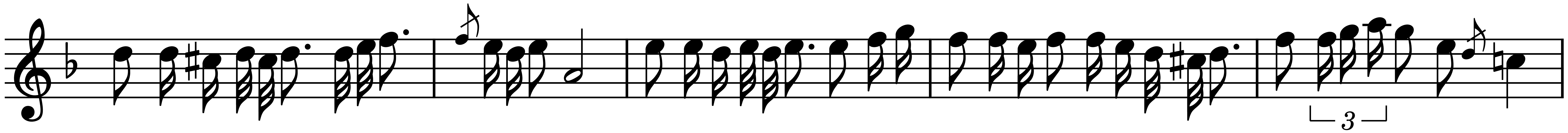


Polonaise in D minor from Notebook for Anna Magdalena Bach BWV Anh II 128

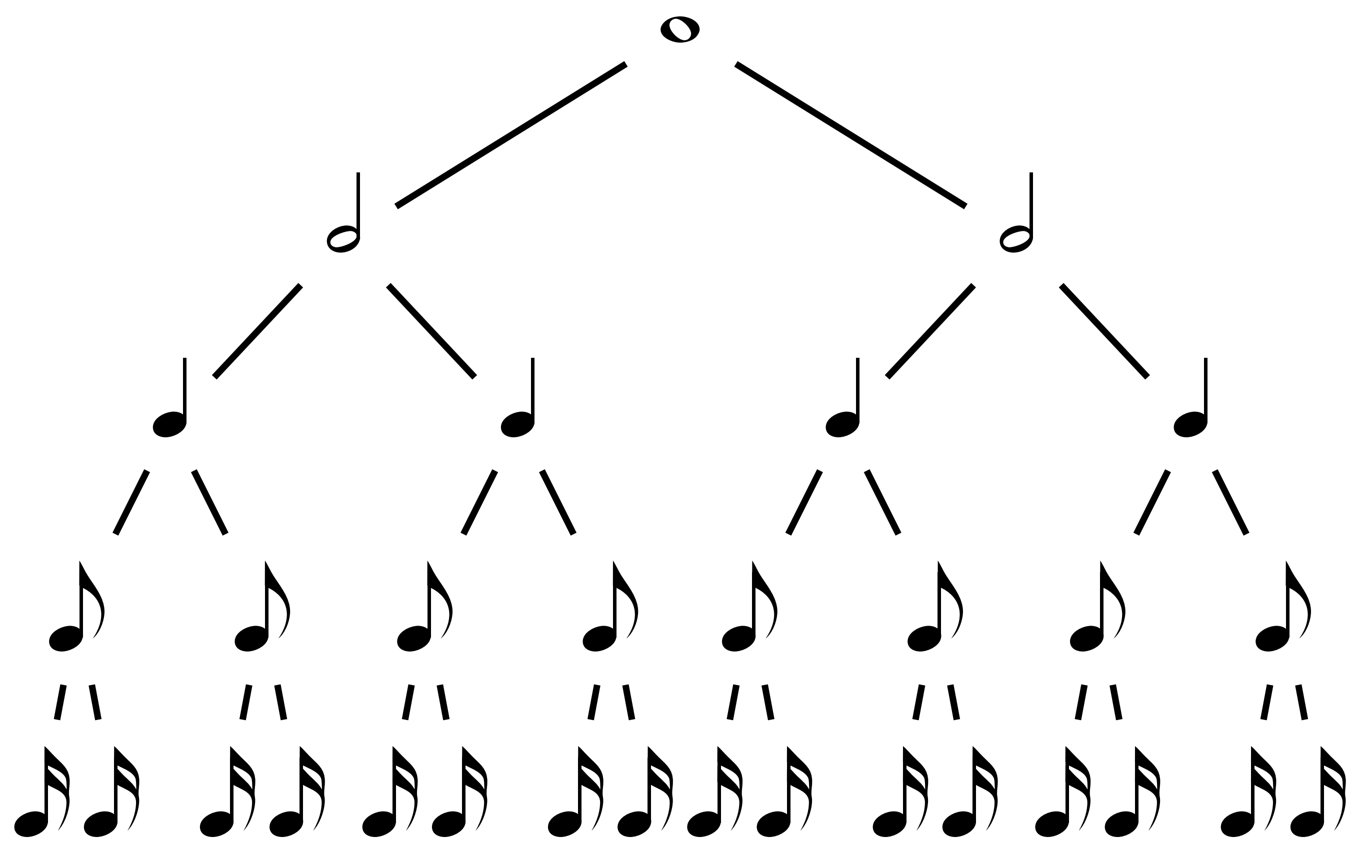
beamed



unbeamed



hierarchical
note
durations



Common Western Music Notation

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

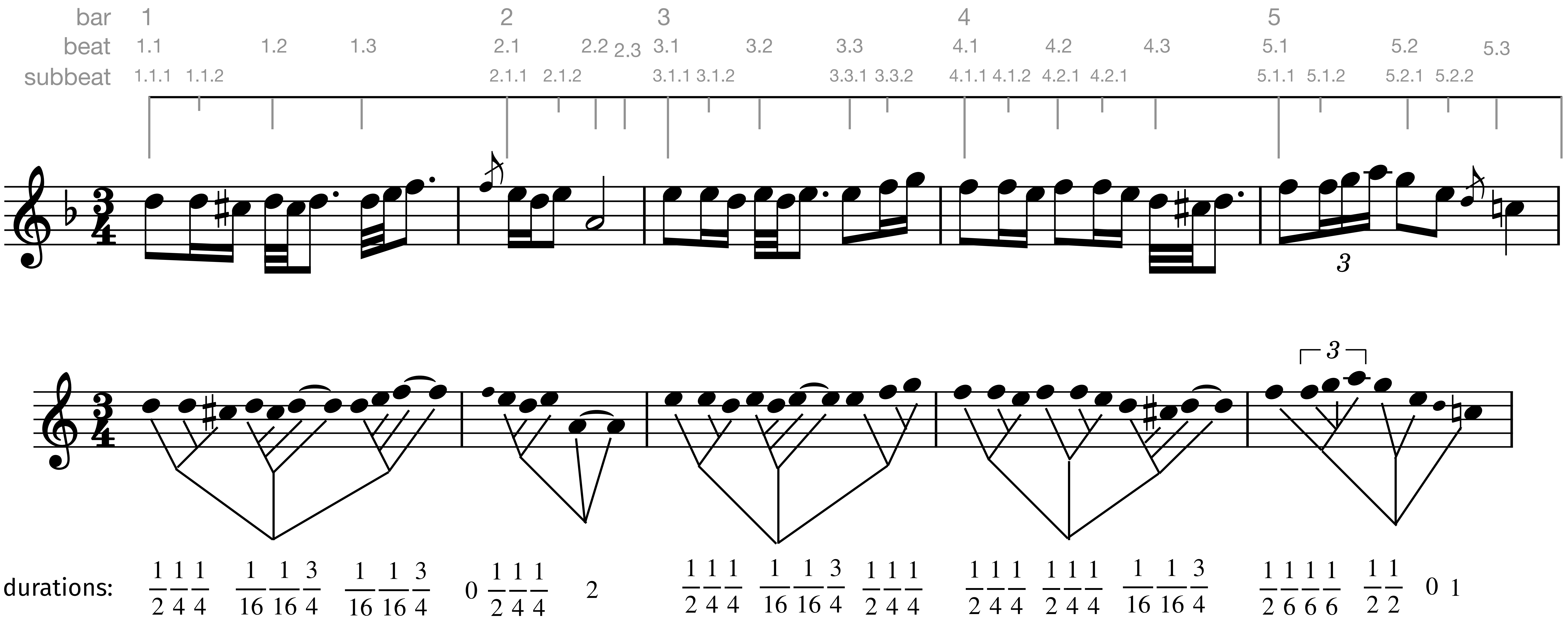
grouping notes with measure bars and beams

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

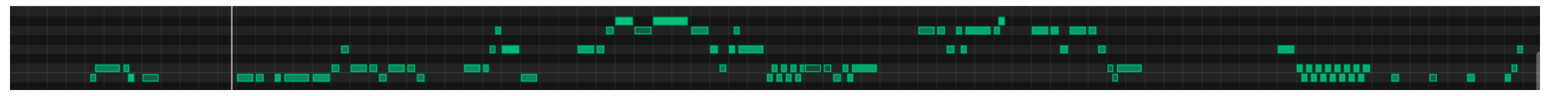
Common Western Music Notation

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

metric structure



music notation = tree-structure, not linear structure piano roll



sequence of timestamped input events

structuring a linear representation according to a language model = **parsing**

leaves = sequence of output events

tree $t \in$ Lang($\mathscr{G}$)

tree-structured representation
of an output music score

conform to a prior language
(expected notation)
defined by a Regular Tree Grammar $\mathscr{G}$

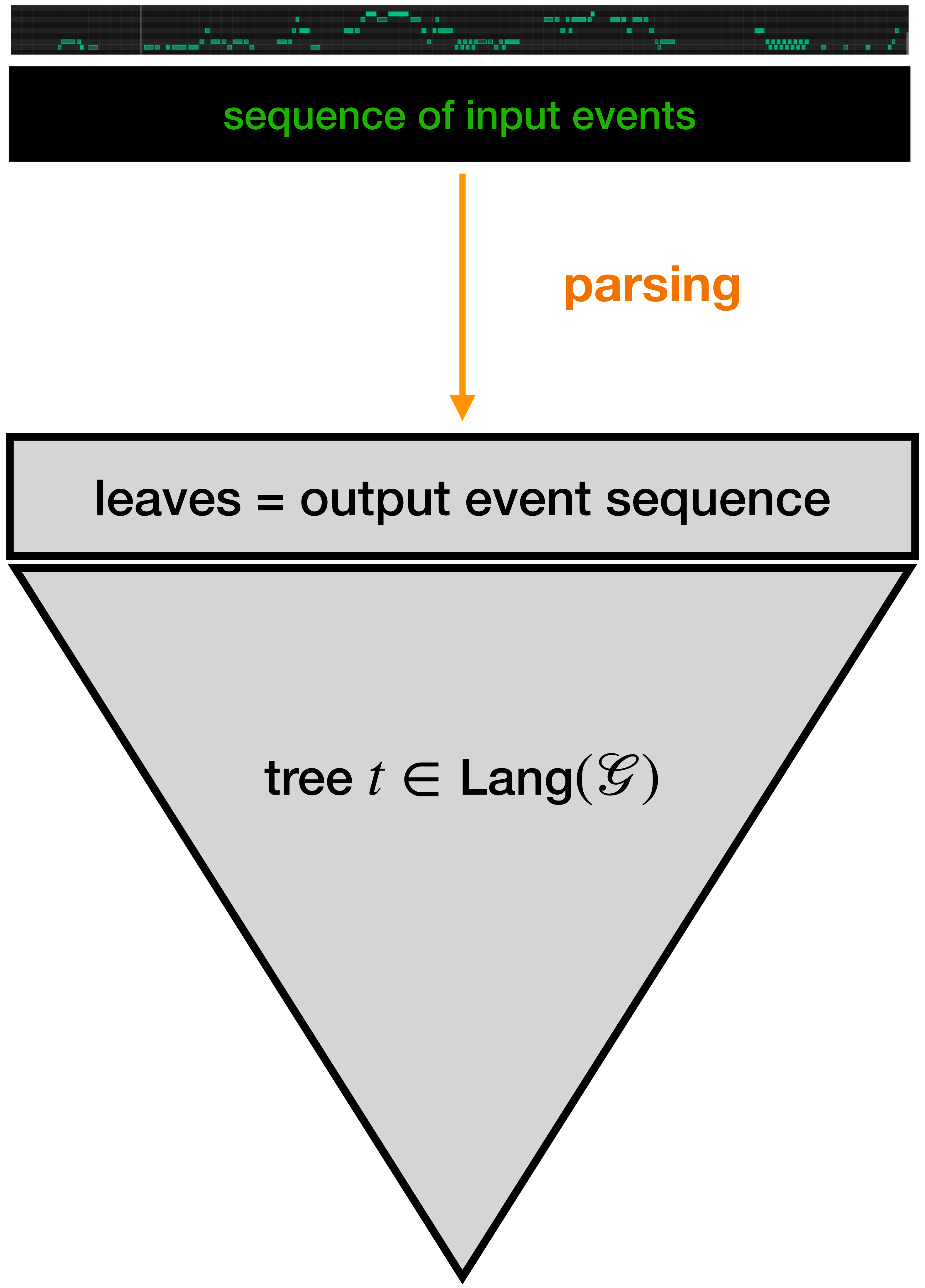
two extensions of parsing are needed for the case music transcription:

1. weighted extension
   a. find best tree
      when prior grammar is ambiguous
   b. input / output measure

2. symbolic extension
   - infinite alphabet

Music Transcription as Weighted Parsing (extension 1.a)

piano roll



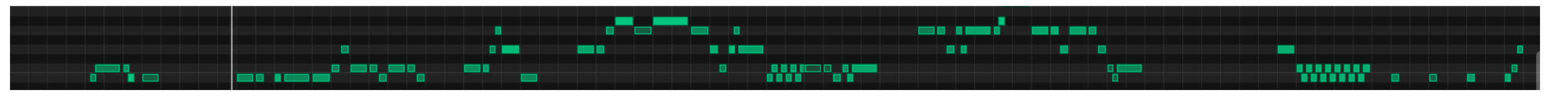
Decision problem:
does there exists a tree t
in the language of $\mathscr{G}$ such that
the leave sequence of t yields
the input event sequence ?

tree-structured representation
of an output music score

conform to a prior language
(expected notation)
defined by a Regular Tree Grammar $\mathscr{G}$

Music Transcription as Weighted Parsing (extension 1.a)

piano roll



sequence of input events

weighted parsing

leaves = output event sequence

tree t of best weight w.r.t. $\mathscr{G}$

tree-structured representation of an output music score

conform to a prior language (expected notation) defined by a Regular Tree Grammar $\mathscr{G}$

Joshua Goodman
Semiring Parsing, Comp. Linguistics, 1999

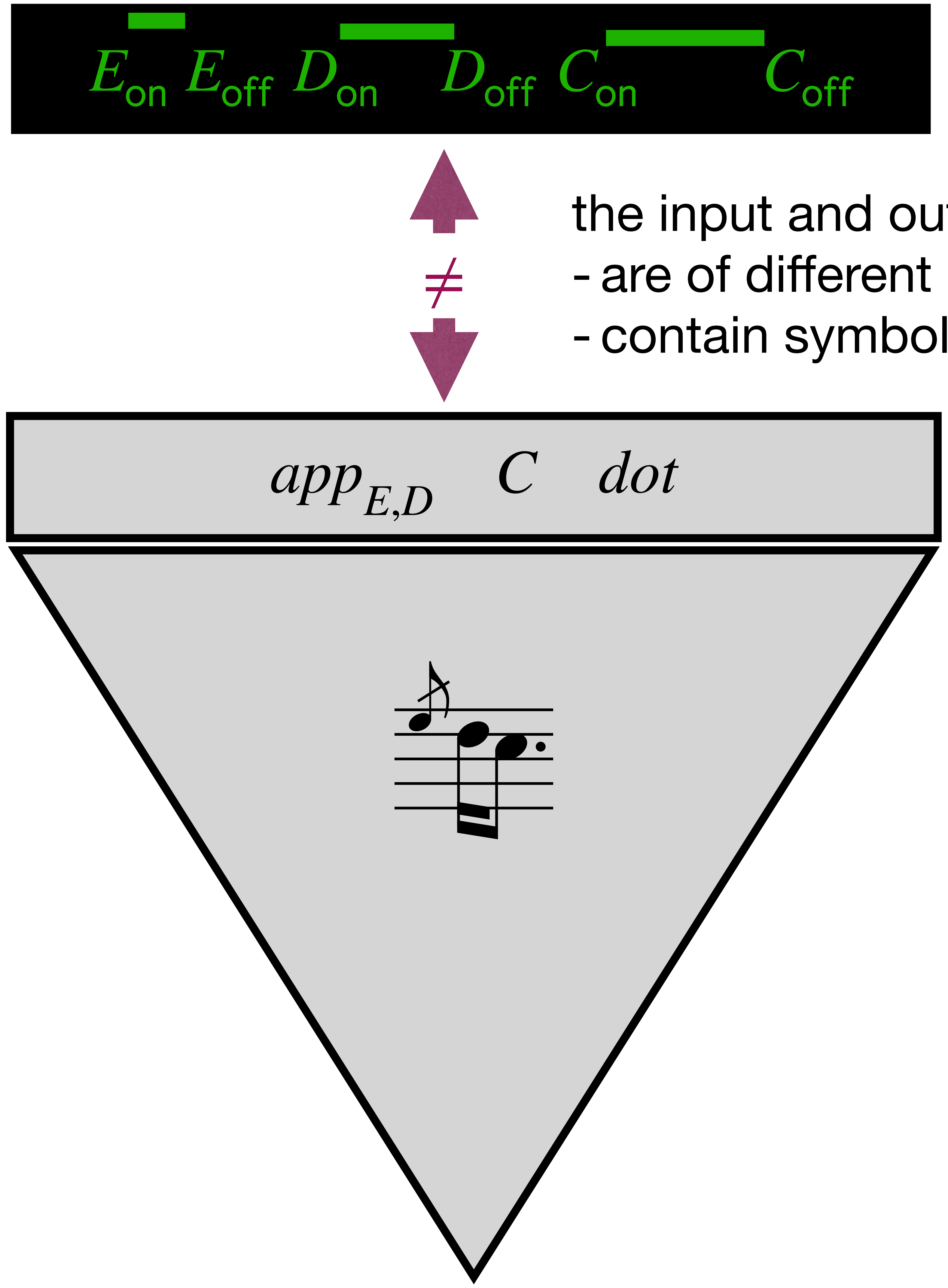
Richard Mörbitz, Heiko Vogler
Weighted Parsing, FSM & NLP, 2019

effective construction of t :
there exist several such trees
when the prior grammar $\mathscr{G}$ is ambiguous

In order to choose one best trees,
rank trees according to their weight values,
computed by a Weighted Tree Grammar.

Music Transcription as Weighted Parsing (extension 1.b)

piano roll
= sequence of timestamped input events



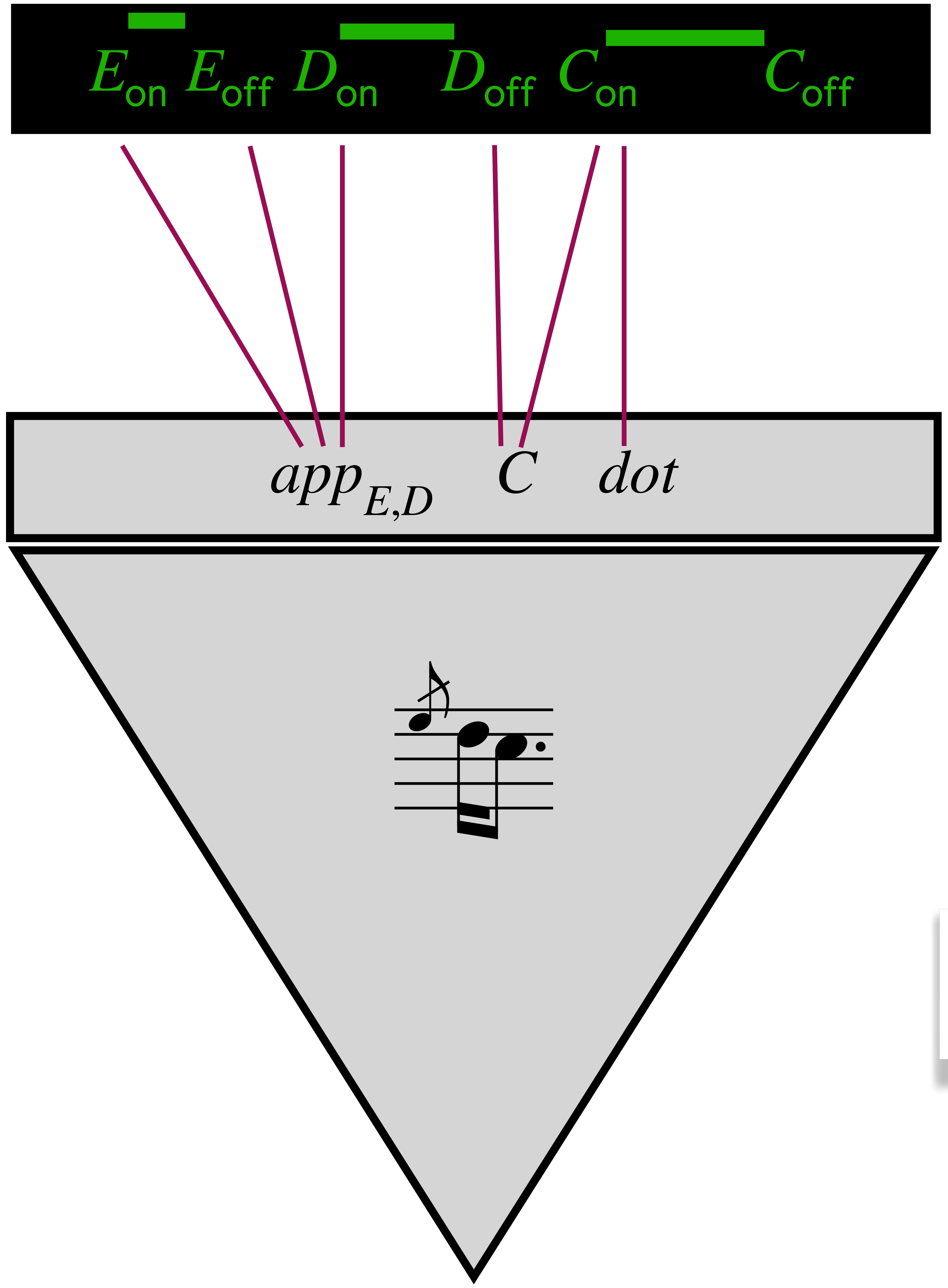
the input and output sequences:
- are of different length,
- contain symbols of different nature.

tree-structured representation
of an output music score

conform to a prior language
(expected notation)
defined by a Regular Tree Grammar $\mathscr{G}$

Music Transcription as Weighted Parsing (extension 1.b)

piano roll
= sequence of timestamped input events



measure of input-output alignement
asynchronous computation by a
Weighted word-to-word Transducer
(stateful definition of an edit-distance)

tree-structured representation
of an output music score

conform to a prior language
(expected notation)
defined by a Regular Tree Grammar $\mathscr{G}$

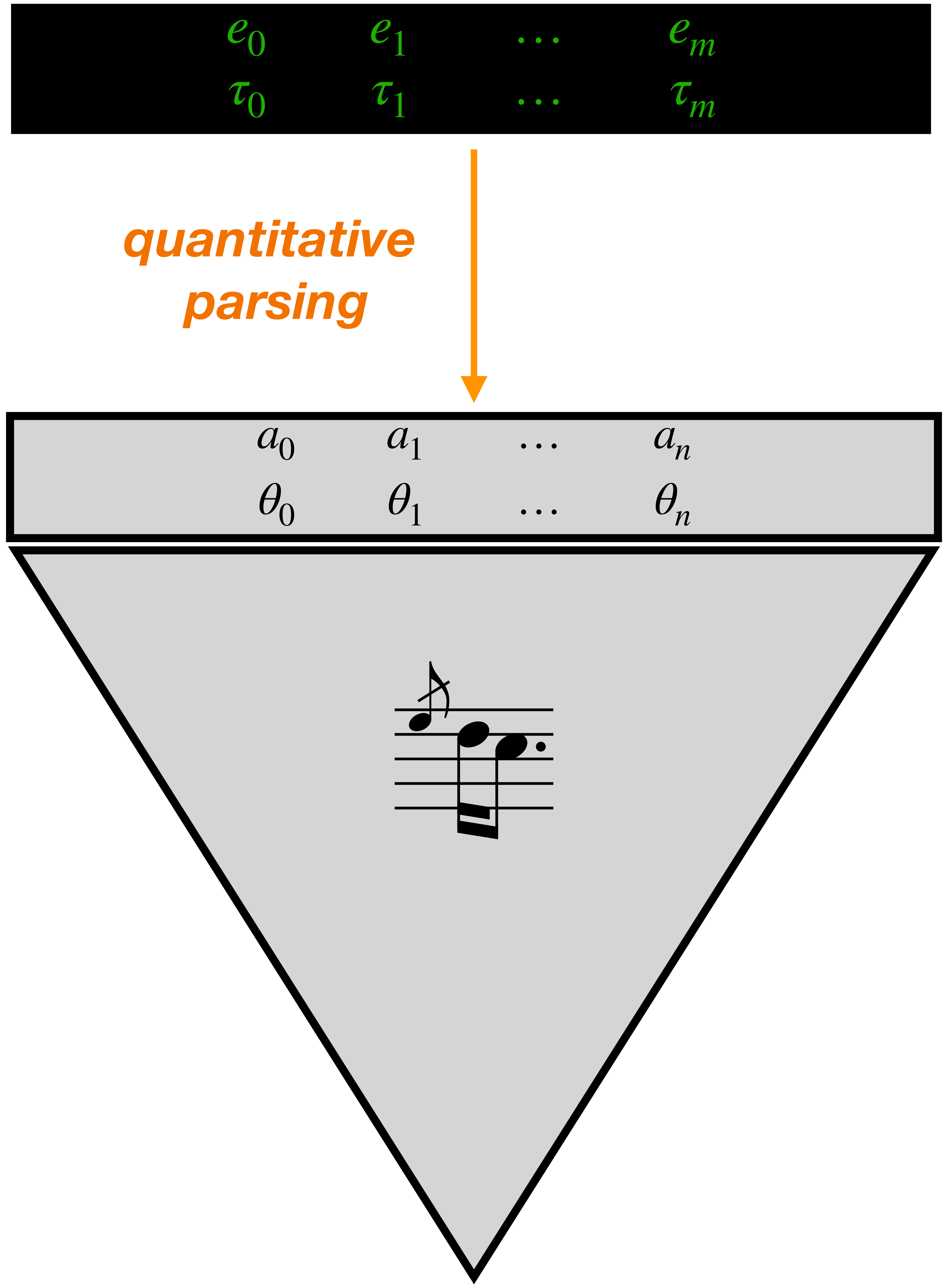
to be combined (for ranking of solutions)
with the weight value computed
by the Weighted Tree Grammar.

Mehryar Mohri
Edit-Distance of Weighted Automata, CIAA 2003

Music Transcription as Quantitative Parsing (extension 2 - Symbolic)

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

piano roll
= sequence of timestamped input events



using Symbolic automata and transducers

Margus Veanes, Loris d'Antoni et al.
CAV 2017, CACM 2021

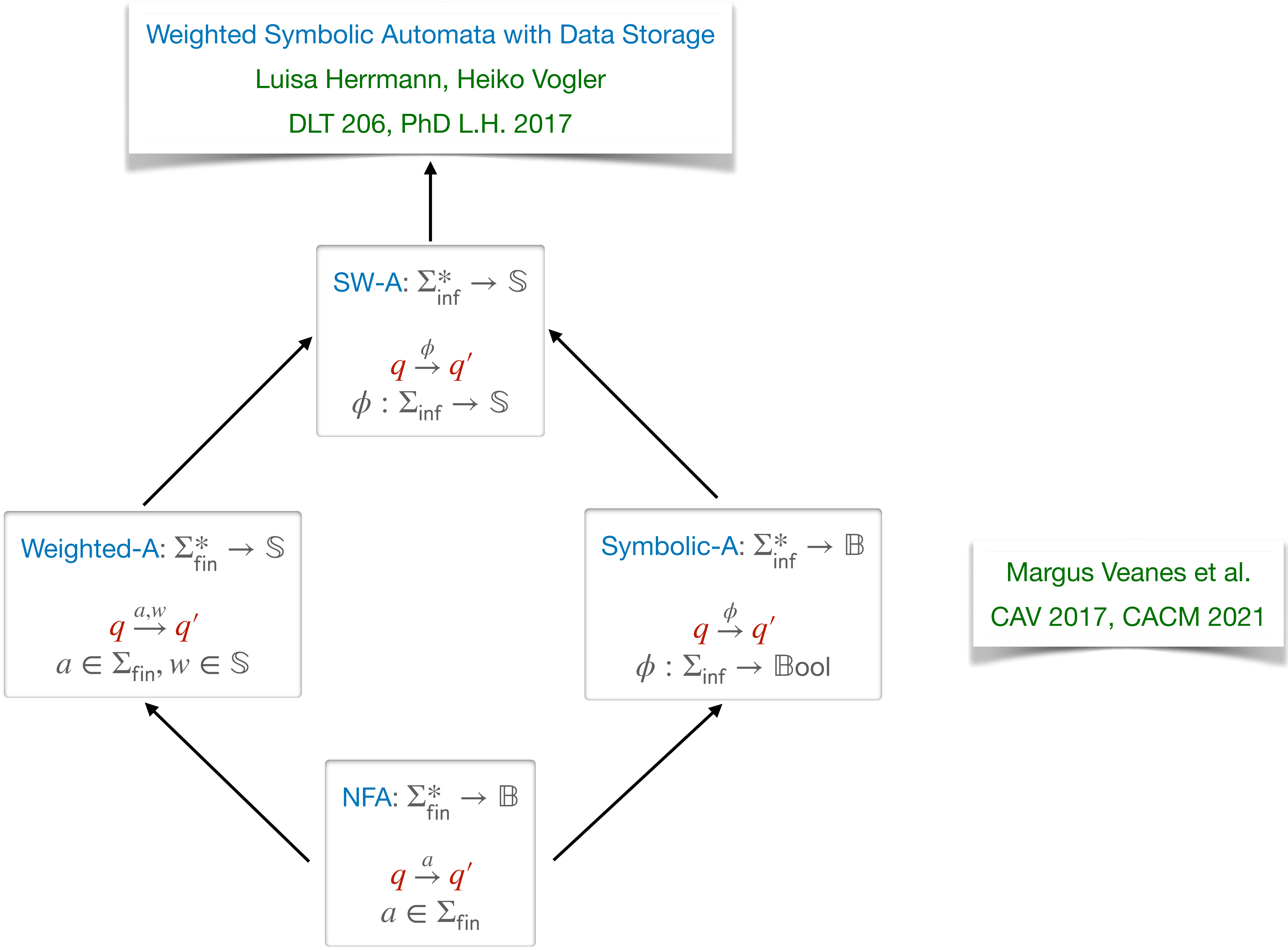
tree-structured representation
of an output music score

conform to a prior language (expected notation)
defined by a Regular Tree Grammar $\mathscr{G}$

Symbolic Weighted Automata

$\mathbb{B}$ is the Boolean algebra
$\mathbb{S}$ is a semiring

Σ_{fin} is a finite alphabet
Σ_{inf} is an infinite alphabet



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

- $\oplus$ is associative and commutative, with neutral element $\mathbb{O}$
- $\otimes$ is associative, with neutral element $\mathbb{I}$
- $\otimes$ 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

$\mathbb{S}$ is:

- commutative if $\otimes$ is commutative
- idempotent if $x \oplus x = x$
- complete if $\oplus$ extends to infinite sums, denoted by $\bigoplus_{i \in I} x_i$ for $I \subseteq \mathbb{N}$

$\bigoplus_{i \in I}$ associative, commutative and $\otimes$ distributes over $\bigoplus_{i \in I}$

Manfred Droste, Werner Kuich, Heiko Vogler
Handbook of Weighted Automata, 2009

Intuitively, $\otimes$ is for the aggregating the weight of the transitions involved in a computation
$\oplus$ is for selecting a best computation

Symbolic Weighted Automata (SW-A)

A SW-A $\mathscr{A}$ over Σ infinite and $\mathbb{S}$ commutative is made of:
- a finite state set Q
- a state entering function in : $Q \to \mathbb{S}$
- a state leaving function out : $Q \to \mathbb{S}$
- a transition function w : $Q \times \Sigma \times Q \to \mathbb{S}$

The weight $\mathscr{A}(s) \in \mathbb{S}$ of a word $s \in \Sigma^*$ is defined by:

$$\text{weight}_{\mathscr{A}}(q, e\,u, q') = \bigoplus_{q'' \in Q} \text{w}(q, e, q'') \otimes \text{weight}_{\mathscr{A}}(q'', u, q')$$

transition step, $e \in \Sigma, u \in \Sigma^*$

$$\text{weight}_{\mathscr{A}}(q, \varepsilon, q) = \mathbb{1}$$

$$\text{weight}_{\mathscr{A}}(q, \varepsilon, q') = \mathbb{0} \text{ if } q \neq q'$$

end-of-computation

$$\mathscr{A}(s) = \bigoplus_{q, q' \in Q} \text{in}(q) \otimes \text{weight}_{\mathscr{A}}(q, s, q') \otimes \text{out}(q')$$

$s \in \Sigma^*$

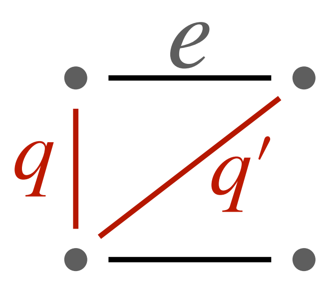
The transition function w is total.
A *missing* transition from q to q' can be specified with $\text{w}(q, a, q') = \mathbb{0}$.

Symbolic Weighted Transducers (SW-T)

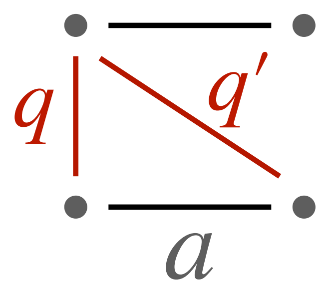
A SW-T $\mathscr{T}$ over Σ (input), Δ (output), and $\mathbb{S}$ commutative is made of:

- a finite state set Q
- a state entering function in : $Q \to \mathbb{S}$
- a state leaving function out : $Q \to \mathbb{S}$
- transition functions

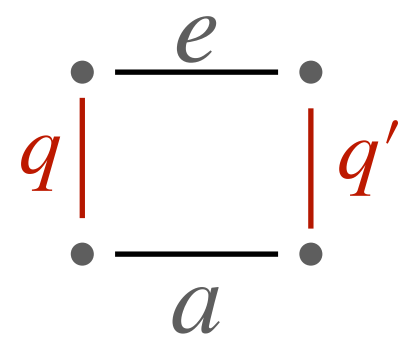
w_{10} : $Q \times \Sigma \times \{\varepsilon\} \times Q \to \mathbb{S}$



w_{01} : $Q \times \{\varepsilon\} \times \Delta \times Q \to \mathbb{S}$



w_{11} : $Q \times \Sigma \times \Delta \times Q \to \mathbb{S}$



The weight $\mathscr{T}(s,t)$ of a pair of words $\langle s,t\rangle \in \Sigma^* \times \Delta^*$ is defined by:

transition: stateful edit-distance

$$\mathrm{weight}_{\mathscr{T}}(q, e\,u, a\,v, q') = \bigoplus_{q''\in Q} \mathrm{w}_{10}(q,e,\varepsilon,q'') \otimes \mathrm{weight}_{\mathscr{T}}(q'', u, a\,v, q') \quad \text{DEL}$$

$$\oplus \bigoplus_{q''\in Q} \mathrm{w}_{01}(q,\varepsilon,a,q'') \otimes \mathrm{weight}_{\mathscr{T}}(q'', e\,u, v, q') \quad \text{INS}$$

$$\oplus \bigoplus_{q''\in Q} \mathrm{w}_{11}(q,e,a,q'') \otimes \mathrm{weight}_{\mathscr{T}}(q'', u, v, q') \quad \text{SUBST}$$

EOC:

$$\mathrm{weight}_{\mathscr{T}}(q, e\,u, \varepsilon, q') = \bigoplus_{q''\in Q} \mathrm{w}_{10}(q,e,\varepsilon,q'') \otimes \mathrm{weight}_{\mathscr{T}}(q'', u, \varepsilon, q')$$

$$\mathrm{weight}_{\mathscr{T}}(q, \varepsilon, a\,v, q') = \bigoplus_{q''\in Q} \mathrm{w}_{01}(q,\varepsilon,a,q'') \otimes \mathrm{weight}_{\mathscr{T}}(q'', \varepsilon, v, q')$$

$$\mathrm{weight}_{\mathscr{T}}(q, \varepsilon, \varepsilon, q) = \mathbb{1}$$

$$\mathrm{weight}_{\mathscr{T}}(q, \varepsilon, \varepsilon, q') = \mathbb{0} \text{ if } q \neq q'$$

$$\mathscr{T}(s,t) = \bigoplus_{q,q'\in Q} \mathrm{in}(q) \otimes \mathrm{weight}_{\mathscr{T}}(q, s, t, q') \otimes \mathrm{out}(q')$$

Over the min-plus semiring

	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

read the ornament (*appoggiatura*):

$$\mathrm{w}_{11}(q_0, e_{\mathsf{on}} \,.\, \tau, app_{e,d} \,.\, \theta, q_d) = |\theta - \tau|$$

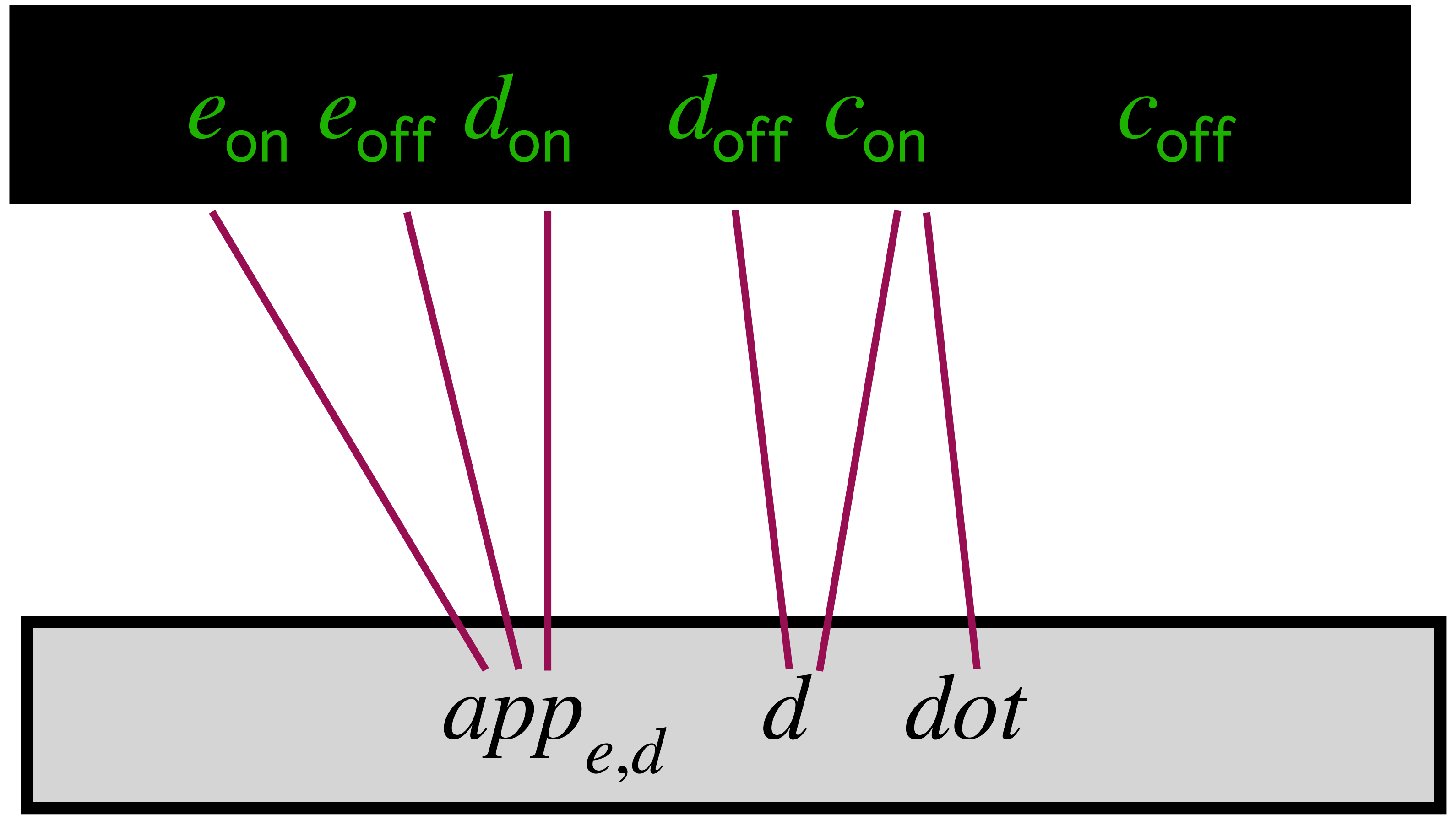
$$\mathrm{w}_{10}(q_d, d_{\mathsf{on}} \,.\, \tau, \varepsilon, q_0) = 0$$

ignore the end of note e (*offset*)

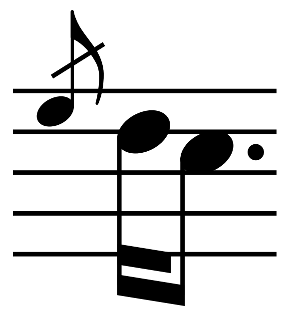
$$\mathrm{w}_{10}(q_0, e_{\mathsf{off}} \,.\, \tau, \varepsilon, q_0) = 0$$

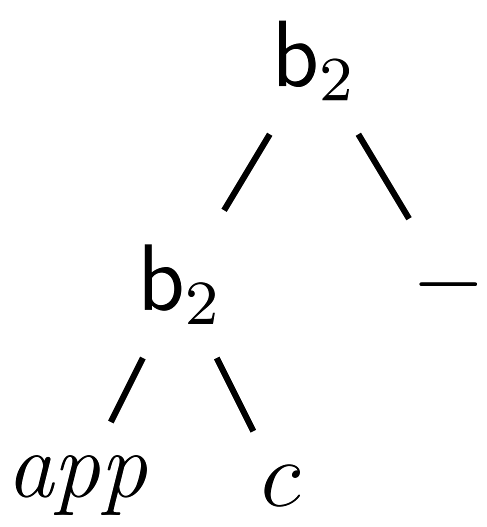
read a note continuation (*tie*)

$$\mathrm{w}_{01}(q_0, \varepsilon, - \,.\theta, q_0) = 0$$



from words to ~~trees~~ structured words





linearised into $[_{\mathsf{b}_2}\, [_{\mathsf{b}_2}\, app\ c\ ]_{\mathsf{b}_2}\ -\ ]_{\mathsf{b}_2}$

call symbols — internal symbols — return symbols

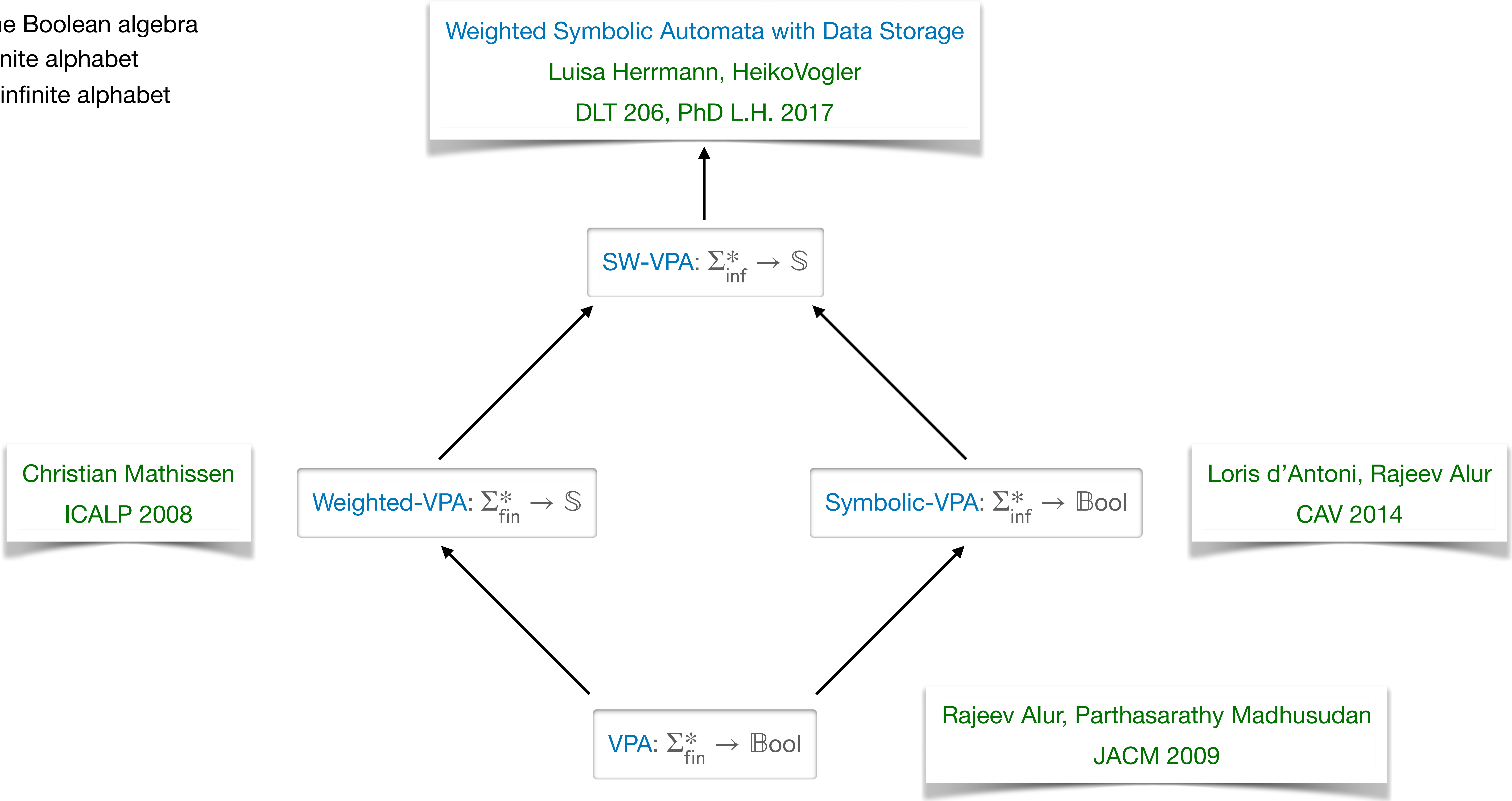
$\Delta = \Delta_c \uplus \Delta_i \uplus \Delta_r$

Rajeev Alur, Parthasarathy Madhusudan

Adding Nesting Structure to Words
JACM 2009

Symbolic Weighted Visibly Pushdown Automata

$\mathbb{B}$ool is the Boolean algebra
Σ_{fin} is a finite alphabet
Σ_{inf} is an infinite alphabet



Symbolic Weighted Visibly Pushdown Automata (SW-VPA)

A SW-VPA $\mathscr{A}$ over $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$ and $\mathbb{S}$ commutative is made of:

- a finite state set Q
- a finite set of stack symbols P
- a state entering function in : $Q \to \mathbb{S}$
- a state leaving function out : $Q \to \mathbb{S}$
- transition functions

$$\mathrm{w_i} : Q \times \Delta_i \times Q \to \mathbb{S}$$

$$\mathrm{w_c} : Q \times \Delta_c \times Q \times P \to \mathbb{S}$$

$$\mathrm{w_r} : Q \times \underbrace{\Delta_c \times P}_{\text{stack top}} \times \Delta_r \times Q \to \mathbb{S}$$

$$\mathrm{weight}_{\mathscr{A}}\big(q[\gamma], a\, v, q'[\gamma']\big) = \bigoplus_{q'' \in Q} \mathrm{w_i}(q, a, q'') \otimes \mathrm{weight}_{\mathscr{A}}\big(q''[\gamma], v, q'[\gamma']\big) \qquad a \in \Delta_i, \; v \in \Delta^*$$

$$\mathrm{weight}_{\mathscr{A}}\big(q[\gamma], c\, v, q'[\gamma']\big) = \bigoplus_{q'' \in Q\, p \in P} \mathrm{w_c}(q, c, q'', p) \otimes \mathrm{weight}_{\mathscr{A}}\big(q''[\langle c, p\rangle\, \gamma], v, q'[\gamma']\big) \qquad c \in \Delta_c$$

$$\mathrm{weight}_{\mathscr{A}}\big(q[\langle c, p\rangle\, \gamma], r\, v, q'[\gamma']\big) = \bigoplus_{q'' \in Q} \mathrm{w_r}\big(q, \underbrace{c, p}_{\text{stack top}}, r, q''\big) \otimes \mathrm{weight}_{\mathscr{A}}\big(q''[\gamma], v, q'[\gamma']\big) \qquad r \in \Delta_r$$

the return transition read both the a top stack and an input symbol.

$$\left.\begin{array}{l} \mathrm{weight}_{\mathscr{A}}\big(q[\gamma], \varepsilon, \varepsilon, q[\gamma]\big) = \mathbb{1} \\ \mathrm{weight}_{\mathscr{A}}\big(q[\gamma], \varepsilon, \varepsilon, q'[\gamma']\big) = \mathbb{0} \text{ if } q \neq q' \text{ or } \gamma \neq \gamma' \end{array}\right\} \text{end-of-computation}$$

$$\mathscr{A}(t) = \bigoplus_{q, q' \in Q} \mathrm{in}(q) \otimes \mathrm{weight}_{\mathscr{A}}\big(q, t, q'\big) \otimes \mathrm{out}(q')$$

Loris d'Antoni, Rajeev Alur
CAV 2014

Symbolic Weighted Visibly Pushdown Transducers (SW-VPT)

A SW-VPT $\mathscr{T}$ over Σ (input),
$\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$ (output),
and $\mathbb{S}$ commutative is made of:

- a finite state set Q
- a state entering function $\mathsf{in} : Q \to \mathbb{S}$
- a state leaving function $\mathsf{out} : Q \to \mathbb{S}$
- transition functions

$$\mathsf{w}_{10} : Q \times \Sigma \times \{\varepsilon\} \times Q \to \mathbb{S}$$
$$\mathsf{w}_{01} : Q \times \{\varepsilon\} \times \Delta_i \times Q \to \mathbb{S}$$
$$\mathsf{w}_{11} : Q \times \Sigma \times \Delta_i \times Q \to \mathbb{S}$$
$$\mathsf{w}_{c} : Q \times \{\varepsilon\} \times \Delta_c \times Q \times P \to \mathbb{S}$$
$$\mathsf{w}_{r} : \underbrace{Q \times \Delta_c \times P}_{\text{stack}} \times \underbrace{\{\varepsilon\}}_{\text{input}} \times \underbrace{\Delta_r}_{\text{output}} \times Q \to \mathbb{S}$$

$$\begin{aligned}\mathsf{weight}_{\mathscr{T}}\big(q[\gamma], e\,u, a\,v, q'[\gamma']\big) = & \bigoplus_{q''\in Q} \mathsf{w}_{10}(q, e, \varepsilon, q'') \otimes \mathsf{weight}_{\mathscr{T}}\big(q''[\gamma], u, a\,v, q'[\gamma']\big)\\ \oplus & \bigoplus_{q''\in Q} \mathsf{w}_{01}(q, \varepsilon, a, q'') \otimes \mathsf{weight}_{\mathscr{T}}\big(q''[\gamma], e\,u, v, q'[\gamma']\big)\\ \oplus & \bigoplus_{q''\in Q} \mathsf{w}_{11}(q, e, a, q'') \otimes \mathsf{weight}_{\mathscr{T}}\big(q''[\gamma], u, v, q'[\gamma']\big)\end{aligned} \quad \left.\begin{matrix} e \in \Sigma \\ u \in \Sigma^* \\ a \in \Delta_i \\ v \in \Delta^*\end{matrix}\right.$$

$$\mathsf{weight}_{\mathscr{T}}\big(q[\gamma], u, c\,v, q'[\gamma']\big) = \bigoplus_{q''\in Q\, p\in P} \mathsf{w}_c(q, \varepsilon, c, q'', p) \otimes \mathsf{weight}_{\mathscr{T}}\big(q''[\langle c, p\rangle\gamma], u, v, q'[\gamma']\big) \quad c \in \Delta_c$$

$$\mathsf{weight}_{\mathscr{T}}\big(q[\langle c, p\rangle\gamma], u, r\,v, q'[\gamma']\big) = \bigoplus_{q''\in Q} \mathsf{w}_r\big(q, c, p, \varepsilon, r, q''\big) \otimes \mathsf{weight}_{\mathscr{T}}\big(q''[\gamma], u, v, q'[\gamma']\big) \quad r \in \Delta_r$$

$$\left.\begin{aligned}\mathsf{weight}_{\mathscr{T}}\big(q[\gamma], e\,u, \varepsilon, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{10}(q, e, \varepsilon, q'') \otimes \mathsf{weight}_{\mathscr{T}}\big(q''[\gamma], u, \varepsilon, q'[\gamma']\big)\\ \mathsf{weight}_{\mathscr{T}}\big(q[\gamma], \varepsilon, a\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{01}(q, \varepsilon, a, q'') \otimes \mathsf{weight}_{\mathscr{T}}\big(q''[\gamma], \varepsilon, v, q'[\gamma']\big)\\ \mathsf{weight}_{\mathscr{T}}\big(q[\gamma], \varepsilon, \varepsilon, q[\gamma]\big) &= \mathbb{1}\\ \mathsf{weight}_{\mathscr{T}}\big(q[\gamma], \varepsilon, \varepsilon, q'[\gamma']\big) &= \mathbb{0} \text{ if } q \neq q' \text{ or } \gamma \neq \gamma'\end{aligned}\right\} \text{EOC}$$

$$\mathscr{T}(s, t) = \bigoplus_{q, q'\in Q} \mathsf{in}(q) \otimes \mathsf{weight}_{\mathscr{T}}\big(q, s, t, q'\big) \otimes \mathsf{out}(q')$$

Th.1: Given a SW-VPT $\mathscr{T}$ over Σ, $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$, and commutative $\mathbb{S}$,
and a SW-VPA $\mathscr{A}$ over Δ, and $\mathbb{S}$,
one can construct in PTIME a SW-VPT $\mathscr{T} \otimes \mathscr{A}$ over Σ, Δ , and $\mathbb{S}$ such that
$\forall s \in \Sigma^* \; (\mathscr{T} \otimes \mathscr{A})(s,t) = \mathscr{T}(s,t) \otimes \mathscr{A}(t)$.

proof:
Cartesian product construction to simulate sychroized computations of $\mathscr{T}$ and $\mathscr{A}$:

$$\mathsf{w}'_{10}\big(\langle q_\mathscr{T}, q_\mathscr{A}\rangle, e, \varepsilon, \langle q'_\mathscr{T}, q_\mathscr{A}\rangle\big) = \mathsf{w}_{10}(q_\mathscr{T}, e, \varepsilon, q'_\mathscr{T})$$

$$\mathsf{w}'_{10}\big(\langle q_\mathscr{T}, q_\mathscr{A}\rangle, e, \varepsilon, \langle q'_\mathscr{T}, q'_\mathscr{A}\rangle\big) = \mathbb{O} \qquad \text{if } q_\mathscr{A} \neq q'_\mathscr{A}$$

$$\mathsf{w}'_{01}\big(\langle q_\mathscr{T}, q_\mathscr{A}\rangle, \varepsilon, a, \langle q_\mathscr{T}, q'_\mathscr{A}\rangle\big) = \mathsf{w}_\mathsf{i}(q_\mathscr{A}, a, q'_\mathscr{A})$$

$$\mathsf{w}'_{01}\big(\langle q_\mathscr{T}, q_\mathscr{A}\rangle, \varepsilon, a, \langle q'_\mathscr{T}, q'_\mathscr{A}\rangle\big) = \mathbb{O} \qquad \text{if } q_\mathscr{T} \neq q'_\mathscr{T}$$

$$\mathsf{w}'_{11}\big(\langle q_\mathscr{T}, q_\mathscr{A}\rangle, e, a, \langle q'_\mathscr{T}, q'_\mathscr{A}\rangle\big) = \mathsf{w}_{11}(q_\mathscr{T}, e, a, q'_\mathscr{T}) \otimes \mathsf{w}_\mathsf{i}(q_\mathscr{A}, a, q'_\mathscr{A})$$

Partial Application of a SW-VPT

Th.2: Given a SW-VPT $\mathscr{T}$ over Σ, $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$, and commutative, complete, idempotent $\mathbb{S}$, and $s \in \Sigma$,
one can construct in PTIME a SW-VPA $\mathscr{T}_s$ over Δ and $\mathbb{S}$ such that $\forall t \in \Delta^* \; \mathscr{T}_s(t) = \mathscr{T}(s, t)$.

proof:

1. extension of SW-VPA with ε-transitions: $\mathsf{w}_{00} : Q \times Q \to \mathbb{S}$.

2. construction of a SW-VPA $\mathscr{T}_s^\varepsilon$ with ε-transitions such that $\forall t \in \Delta^* \; \mathscr{T}_s^\varepsilon(t) = \mathscr{T}(s, t)$

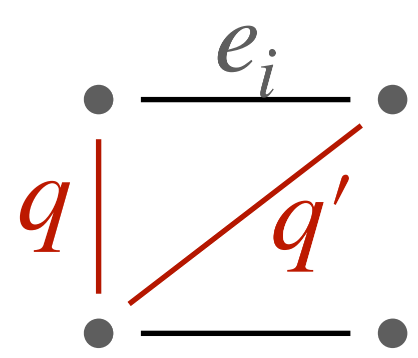
let $\mathscr{T} = \langle Q, P, \mathsf{in}, \mathsf{w}_{10}, \mathsf{w}_{01}, \mathsf{w}_{11}, \mathsf{w}_c, \mathsf{w}_r, \mathsf{out} \rangle$ and $s = e_1 \ldots e_k$. $Q' = [0..k] \times Q$, transition functions of $\mathscr{T}_s$:

$$\mathsf{w}_i'\big(\langle i, q\rangle, a, \langle i, q'\rangle\big) = \mathsf{w}_{01}(q, \varepsilon, a, q')$$

$$\mathsf{w}_i'\big(\langle i, q\rangle, a, \langle i+1, q'\rangle\big) = \mathsf{w}_{11}(q, e_i, a, q')$$

$$\mathsf{w}_{00}'\big(\langle i, q\rangle, \langle i+1, q'\rangle\big) = \mathsf{w}_{10}(q, e_i, \varepsilon, q')$$

introducing ε-transition



$$\mathsf{w}_c'\big(\langle i, q\rangle, c, \langle i, q'\rangle, p\big) = \mathsf{w}_c(q, \varepsilon, c, q', p), \quad \mathsf{w}_r'\big(\langle i, q\rangle, c, p, r, \langle i, q'\rangle\big) = \mathsf{w}_r(q, c, p, \varepsilon, r, q')$$

3. removal of ε-transitions: construction of a SW-VPA $\mathscr{T}_s$ such that $\forall t \in \Delta^* \; \mathscr{T}_s(t) = \mathscr{T}_s^\varepsilon(t)$

Th.2: Given a SW-VPT $\mathscr{T}$ over Σ, $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$, and commutative, complete, idempotent $\mathbb{S}$, and $s \in \Sigma$,
one can construct in PTIME a SW-VPA $\mathscr{T}_s$ over Δ and $\mathbb{S}$ such that $\forall t \in \Delta^* \; \mathscr{T}_s(t) = \mathscr{T}(s, t)$.

proof:

…

3. removal of ε-transitions: construction of a SW-VPA $\mathscr{T}_s$ such that $\forall t \in \Delta^* \; \mathscr{T}_s(t) = \mathscr{T}_s^{\varepsilon}(t)$

precompute ε-sequences $\ell(q, q') = \bigoplus_{\substack{q_0=q \; q_n=q' \\ q_0 \ldots q_n \in Q^*}} \bigotimes_{i=0}^{n-1} \mathrm{w}_{00}(q_i, q_{i+1})$ without repetition by idempotency of $\mathbb{S}$,

and complete transition functions with them.

Searching for a minimal witness for a SW-VPA

Th.3: Given a effective SW-VPA $\mathscr{A}$ over $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$, and commutative, complete, idempotent, total and bounded $\mathbb{S}$, one can construct in PTIME $t \in \Delta^*$ such that $\mathscr{A}(t) = \bigoplus_{v \in \Delta} \mathscr{T}(v)$.

The restrictions

$\mathbb{S}$ is assumed :

- idempotent $x \oplus x = x$ It 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.

	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

Searching for a minimal witness for a SW-VPA

Th.3: Given a effective SW-VPA $\mathscr{A}$ over $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$, and commutative, complete, idempotent, total and bounded $\mathbb{S}$, one can construct in PTIME $t \in \Delta^*$ such that $\mathscr{A}(t) = \bigoplus_{v \in \Delta} \mathscr{T}(v)$.

The restrictions

$\mathscr{A}$ is assumed effective: for all $q, q' \in Q$
the global minimum $\bigoplus_{a \in \Delta_i} \mathrm{w_i}(q, a, q')$ is known, as well as one $a_0 \in \Delta_i$ making $\mathrm{w_i}(q, a_0, q')$ reach this minimum.

Similarly for $\bigoplus_{c \in \Delta_c} \mathrm{w_c}(q, c, q', p)$ and $\bigoplus_{c \in \Delta_c} \bigoplus_{r \in \Delta_r} \mathrm{w_r}(q, c, p, r, q')$.

Th.3: Given a effective SW-VPA $\mathscr{A}$ over $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$, and commutative, complete, idempotent, total and bounded $\mathbb{S}$, one can construct in PTIME $t \in \Delta^*$ such that $\mathscr{A}(t) = \bigoplus_{v \in \Delta} \mathscr{T}(v)$.

Let $\mathscr{A} = \langle Q, P, \mathsf{in}, \mathsf{w_i}, \mathsf{w_c}, \mathsf{w_r}, \mathsf{out} \rangle$. We build a weighted graph $\mathscr{G}(\mathscr{A})$ with vertices of the form

- $\langle q, \bot, q' \rangle$: computations starting in state q with empty stack and ending in state q' with empty stack.
- $\langle q, \top, q' \rangle$: computations starting in state q with a non-empty stack γ and ending in state q' with the same stack γ.

and weighted edges expressing the extension of the represented computations:

- $\langle q_1, \bot, q_2 \rangle \xrightarrow{\bigoplus_{a \in \Delta_i} (\mathsf{w_i}(q_1, a, q_2))} \langle q_0, \bot, q_3 \rangle$ the computation is extended with one internal transition step.

- $\langle q_1, \top, q_2 \rangle \xrightarrow{\bigoplus_{p \in P} \bigoplus_{c \in \Delta_c} (\mathsf{w_c}(q_0, c, q_1, p) \otimes \bigoplus_{r \in \Delta_r} \mathsf{w_r}(q_2, c, p, r, q_3))} \langle q_0, \bot, q_3 \rangle$
  the computation is extended with one call step on the left and one return oe the right.

- $\langle q_1, \top, q_2 \rangle \xrightarrow{\bigoplus_{p \in P} \bigoplus_{c \in \Delta_c} (\mathsf{w_c}(q_0, c, q_1, p) \otimes \bigoplus_{r \in \Delta_r} \mathsf{w_r}(q_2, c, p, r, q_3))} \langle q_0, \top, q_3 \rangle$.

For all $q, q' \in Q$ search for a best path in $\mathscr{G}(\mathscr{A})$ from some $\langle q_0, \bot, q_0 \rangle$ or $\langle q_0, \top, q_0 \rangle$ and ending with $\langle q, \bot, q' \rangle$.

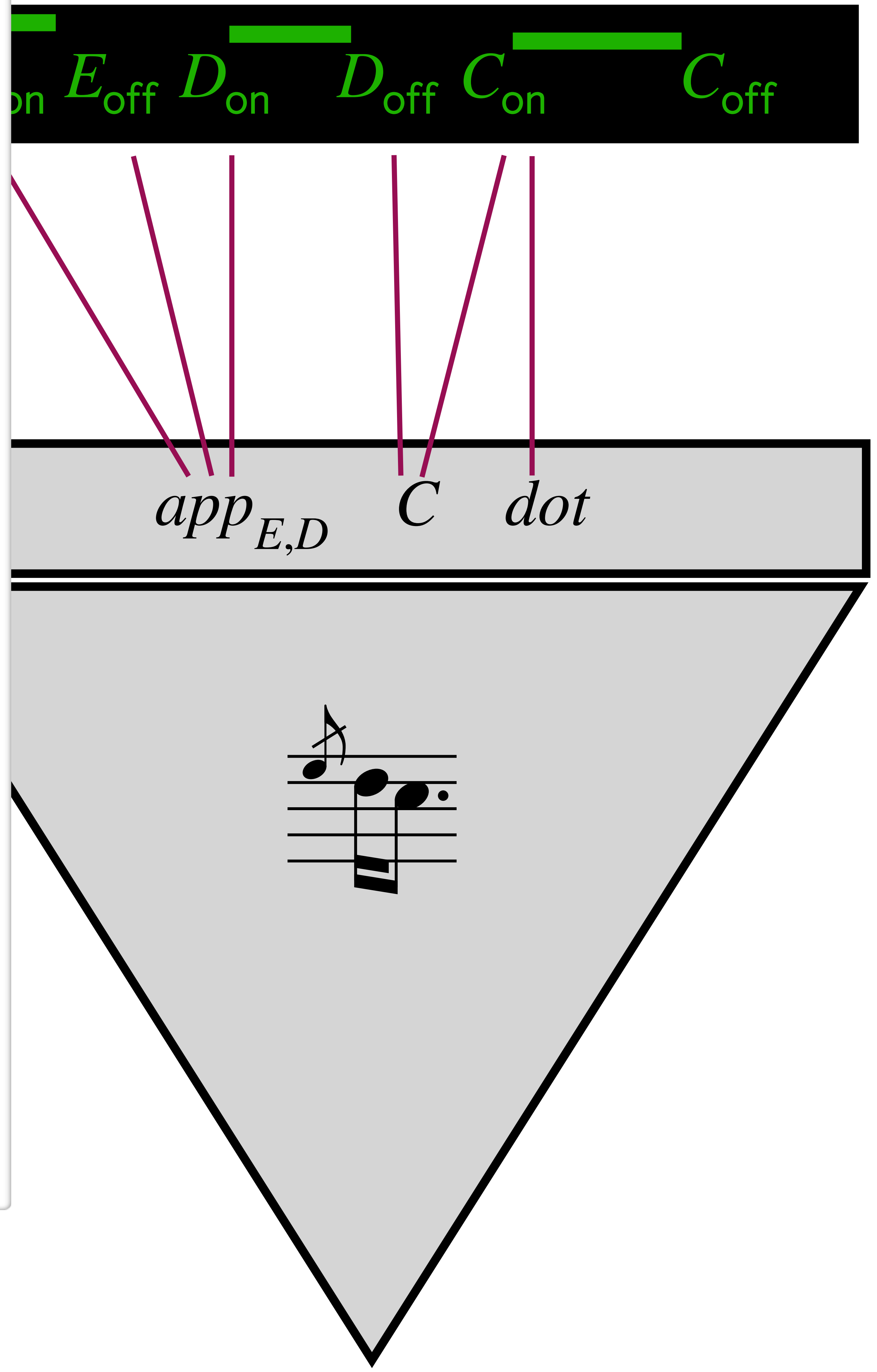
Quantitative Parsing

Assuming:
- Σ input alphabet
- $\Delta = \Delta_c \uplus \Delta_r \uplus \Delta_i$
- $\mathbb{S}$ commutative

Given:
- a SW-T $\mathcal{T}$ over Σ, Δ, and $\mathbb{S}$
  $\mathcal{T} : \Sigma^* \times \Delta^* \to \mathbb{S}$
- a SW-VPA $\mathcal{A}$ over Δ, and $\mathbb{S}$
  $\mathcal{A} : \Delta^* \to \mathbb{S}$
- an unstructured input word $s \in \Sigma^*$

Find a ~~tree~~ structured output word $t \in \Delta^*$ s.t.

$$\mathcal{T}(s,t) \otimes \mathcal{A}(t) = \bigoplus_{v \in \Delta^*} \mathcal{T}(s,v) \otimes \mathcal{A}(v)$$



measure of input-output alignement computed by the Weighted word-to-word Transducer

$\mathcal{T}(s,t)$

$\otimes$

$\mathcal{A}(t)$

weight value computed by the Weighted Tree Grammar

Mehryar Mohri
Edit-Distance of Weighted Automata, CIAA 2003

Automated Music Transcription: qparse framework

$$\mathscr{T}(s,t) \otimes \mathscr{A}(t) = \bigoplus_{v \in \Delta^*} \mathscr{T}(s,v) \otimes \mathscr{A}(v)$$

- C++ implementation (75k loc)
- command line tools
- Python binding
- online port, real-time processing

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

Monophonic transcription: evaluation on a method for learning rhythm

monophonic : one note at a time
Good results for complex cases (ornaments, mixed tuplets, mixed note durations, silences…)
~ 100ms for the transcription of 1 score

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

original score

transcription of MIDI recording by qparse

Monophonic transcription: evaluation on a method for learning rhythm

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

original score

transcription of MIDI recording by Finale

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-to-MIDI transcription procedure
- prior beat (measure) tracking

Drum 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

- closure results and 1-best parsing algorithm for classes of Symbolic Weighted (Visibly Pushdown) Automata ad Transducers
- application to parsing i presence of infinite alphabets
- based on prior transducer computed distance between the input word and yield sequence

- implemented within a C++ library for Automated Music Transcription
- ongoing case studies on transcription of monophonic and polyphonic (drums, piano) collections

- extension of the 1-best theorem 3 to n-best
  extension of this theorem from SW-VPA to SW-VPT

THANK YOU FOR YOUR ATTENTION!

"I'll be Bach"