Symbolic Weighted Language Models, Quantitative Parsing and Automated Music Transcription

Florent Jacquemard, Lydia Rodriguez de La Nava

Symbolic Weighted Language Models, Quantitative Parsing and Automated Music Transcription

Florent Jacquemard and Lydia Rodriguez-de la Nava

INRIA and CNAM/Cedric lab, Paris, France

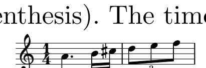
Abstract. We study several classes of symbolic weighted formalisms: automata (swA), transducers (swT) and visibly pushdown extensions (swVPA, swVPT). They combine the respective extensions of their symbolic and weighted counterparts, allowing a quantitative evaluation of words over a large or infinite input alphabet.
We present properties of closure by composition, the computation of transducer-defined distances between nested words and languages, as well as a PTIME 1-best search algorithm for swVPA. These results are applied to solve in PTIME a variant of parsing over infinite alphabets. We illustrate this approach with a motivating use case in automated music transcription.

1 Introduction

Symbolic Weighted (sw) language models [12] (automata and transducers) combine two important extensions of standard models. On the one hand, symbolic extensions, like *Symbolic Automata* (sA [8]), can handle an infinite input alphabet Σ, by guarding every transition with a predicate $\phi : \Sigma \to \mathbb{B}$. The ability of sA to compare input symbols is quite restricted, compared to other models of automata extended e.g. with registers (see [20] for a survey), however, under appropriate closure conditions on the set of predicates, all the good properties enjoyed by automata over finite alphabets are still valid for sA.
On the other hand, *Weighted Automata* (wA [9]) extend qualitative evaluation of input words to *quantitative* evaluation, by assigning to every transition a weight value in a semiring $\mathbb{S}$. The weights of the rules involved in a computation are combined using the product operator $\otimes$ of $\mathbb{S}$, whereas the sum operator $\oplus$ of $\mathbb{S}$ is used to resolve ambiguity (typically, $\oplus$ selects, amongst two computations, the best weighted one). These extensions have also been applied to evaluate hierarchical structures, like trees [9, ch. 9], or nested words, with symbolic [7], or weighted [6] extensions of Visibly Pushdown Automata (VPA [2]). With their ability to evaluate data sequences quantitatively, sw models have found various applications such as data stream processing [3], runtime verification of timed systems [23] or robustness optimization for machine learning models [15].

The sw models with data storage defined in [12], where their expressiveness is extensively studied, are very general, and cover all the models cited above, as well as those considered in this paper. Here, we consider simple models of sw-automata and transducers whose transitions are assigned *functions* $\phi : \Sigma \to \mathbb{S}$ from input symbols in an infinite alphabet Σ into a semiring $\mathbb{S}$ (Section 3). Such functions generalize the boolean guards of symbolic models, from the Boolean codomain $\mathbb{B}$ to an arbitrary semiring $\mathbb{S}$, and the constant values of weighted models. We prove some properties of closure under composition for those sw models, generalizing classical constructions for the composition of transducers [19], and propose a polynomial time algorithm of search for a word of minimal weight for swVPA (somehow a variant of reachability problems in pushdown automata [5]). We apply these results to the problem of parsing words over infinite alphabet (sw-*parsing*), whose goal is: given an (unstructured) input word s, to find a (structured) nested word t at a minimal distance from s, where the distance, following [17], is defined by $T(s,t) \otimes A(t)$, T being a sw-transducer (swT) and A a swVPA (Section 4). The notion of transducer-based distances allows to consider different infinite alphabets for the input s and output t. Moreover, the use of swVPA allows to search for an output t in the form of a nested word, as a linear representation of a parse tree; *sw-parsing* is solved with a Bar-Hillel, Perles and Shamir construction [11, ch. 13], and the best-search algorithm for swVPA. We illustrate our approach with an application that motivated this work: *automated music transcription*, i.e. the problem of converting a linear music recording given in input into a score in *Common Western Music Notation*, a representation structured hierarchically [24].

Example 1. Let us consider a short input sequence I of musical events represented by symbols of the form $e{:}\,\tau$ in an infinite alphabet Σ, where e is a MIDI key number in 21..108 [21], or the mark 'start' or 'end', and $\tau \in \mathbb{Q}$ is a duration in seconds. Such inputs typically correspond to the recording of a live performance: $I = \mathsf{start}{:}\,0.07, 69{:}\,0.65, 71{:}\,0.19, 73{:}\,0.14, 74{:}\,0.31, 76{:}\,0.35, 77{:}\,0.29, \mathsf{end}{:}\,0.$

The output of parsing I is a nested word O, where separated notes are grouped into hierarchical patterns. It is made of symbols $a{:}\,\tau'$ in an alphabet Δ, where a is either a *note name*, (e.g., A_4, G_5, etc.), a *continuation* symbol '$-$', or a *markup* symbol (opening or closing parenthesis). The time value τ' is a musical duration. For instance, the music score  is represented by the nested word: $O = \lceil_{\mathsf{m}}{:}\,2, \lceil_2{:}\,1, \mathsf{A}_4{:}\,\frac{1}{2}, \lceil_2{:}\,\frac{1}{2}, -{:}\,\frac{1}{4}, \lceil_2{:}\,\frac{1}{4}, \mathsf{B}_4{:}\,\frac{1}{8}, \mathsf{C}\sharp_5{:}\,\frac{1}{8}, \rceil_2{:}\,\frac{1}{4}, \rceil_2{:}\,\frac{1}{2}, \rceil_2{:}\,1, \lceil_{\mathsf{m}}{:}\,1, \lceil_3{:}\,1, \mathsf{D}_5{:}\,\frac{1}{3}, \mathsf{E}_5{:}\,\frac{1}{3}, \mathsf{F}_5{:}\,\frac{1}{3}, \rceil_3{:}\,1, \rceil_{\mathsf{m}}{:}\,1, \rceil_{\mathsf{m}}{:}\,2$ (see Figure 1). The symbol $\lceil_{\mathsf{m}}$ marks the opening of a *measure* (a time interval of duration 1), while the subsequences of O between markups $\lceil_d{:}\,\ell$ and $\rceil_d{:}\,\ell$, for some natural number d, represent the division of a duration ℓ by d. The sequence $O \in \Delta^*$ is a candidate solution for the transcription of $I \in \Sigma^*$. Let us consider another candidate , represented by $O' = \lceil_{\mathsf{m}}{:}\,2, \lceil_2{:}\,1, \mathsf{A}_4{:}\,\frac{1}{2}, \lceil_2{:}\,\frac{1}{2}, -{:}\,\frac{1}{4}, \mathsf{B}_4{:}\,\frac{1}{4}, \rceil_2{:}\,\frac{1}{2}, \rceil_2{:}\,1, \lceil_{\mathsf{m}}{:}\,1, \lceil_3{:}\,1, \text{`}\mathsf{C}\sharp_5\text{'}{:}0, \mathsf{D}_5{:}\,\frac{1}{3}, \mathsf{E}_5{:}\,\frac{1}{3}, \mathsf{F}_5{:}\,\frac{1}{3}, \rceil_3{:}\,1, \rceil_{\mathsf{m}}{:}\,1, \rceil_{\mathsf{m}}{:}\,2$. The quoted symbol '$\mathsf{C}\sharp_5$' represents an *appogiatura*, i.e. an ornamental note of theoretical duration 0. Roughly, sw-parsing associates a weight value to each candidate, depending of its temporal distance to I and notational complexity. Our goal is to find a best candidate. $\diamond$

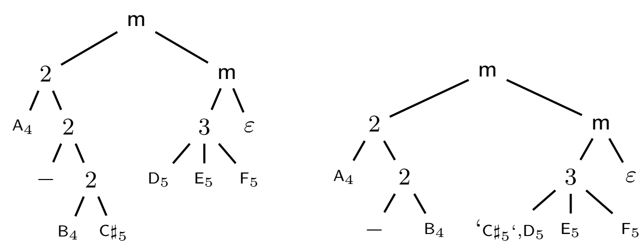


Fig. 1. Tree representation of the scores of Ex 1, linearized respectively into O and O'.

2 Preliminary Notions

Semirings. We shall consider weight values in a *semiring* $\langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$: a structure of domain $\mathbb{S}$, equipped with two associative binary operators $\oplus$ and $\otimes$, with respective neutral elements $\mathbb{0}$ and $\mathbb{1}$, such that $\oplus$ is commutative, $\otimes$ distributes over $\oplus$: $\forall x, y, z \in \mathbb{S}$, $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$, and $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$, $\mathbb{0}$ is absorbing for $\otimes$: $\forall x \in \mathbb{S}$, $\mathbb{0} \otimes x = x \otimes \mathbb{0} = \mathbb{0}$.

A semiring $\mathbb{S}$ is *commutative* if $\otimes$ is commutative. It is *idempotent* if for every $x \in \mathbb{S}$, $x \oplus x = x$. Every idempotent semiring $\mathbb{S}$ induces a partial ordering $\leq_\oplus$ called the *natural ordering* of $\mathbb{S}$ [16], defined by: for every $x, y \in \mathbb{S}$, $x \leq_\oplus y$ iff $x \oplus y = x$. It is sometimes defined in the opposite direction [9, ch. 1]; we follow here the direction that coincides with the usual ordering on the Tropical semiring *min-plus* (Figure 2). An idempotent semiring $\mathbb{S}$ is called *total* if $\leq_\oplus$ is total, i.e. when for every $x, y \in \mathbb{S}$, either $x \oplus y = x$ or $x \oplus y = y$.

Lemma 1 (Monotony, [16]). *If $\langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$ is idempotent, for every $x, y, z \in \mathbb{S}$, $x \leq_\oplus y$ implies $x \oplus z \leq_\oplus y \oplus z$, $x \otimes z \leq_\oplus y \otimes z$ and $z \otimes x \leq_\oplus z \otimes y$.*

We say that $\mathbb{S}$ is *monotonic* wrt $\leq_\oplus$. Another important semiring property in the context of optimization is *superiority* ((i) of Lemma 2), which generalizes the *non-negative weights* condition in Dijkstra's shortest-path algorithm. Intuitively, it means that combining elements with $\otimes$ always increases their weight.

Lemma 2 (Superiority, Boundedness). *Let $\langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$ be an idempotent semiring. The two following statements are equivalent: (i) for all $x, y \in \mathbb{S}$, $x \leq_\oplus x \otimes y$ and $y \leq_\oplus x \otimes y$ (ii) for all $x \in \mathbb{S}$, $\mathbb{1} \oplus x = \mathbb{1}$.*

The property (i) of superiority implies that $\mathbb{1} \leq_\oplus z \leq_\oplus \mathbb{0}$ for all $z \in \mathbb{S}$ (by setting $x = \mathbb{1}$ and $y = \mathbb{0}$ in Lemma 2). From an optimization point of view, it means that $\mathbb{1}$ is the best value, and $\mathbb{0}$ the worst. A semiring $\mathbb{S}$ with property (ii) of Lemma 2 is called *bounded* in [16] and in the rest of the paper.

Lemma 3 ([16], Lemma 3). *Every bounded semiring is idempotent.*

We need to extend $\oplus$ to infinitely many operands. A semiring $\mathbb{S}$ is called *complete* [9, ch. 1] if it has an operation $\bigoplus_{i \in I} x_i$ for every family $(x_i)_{i \in I}$ of elements in the domain of $\mathbb{S}$, over an index set $I \subseteq \mathbb{N}$, such that:

	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	$+$	∞	0

Fig. 2. Some commutative, bounded, total and complete semirings.

i. infinite sums extend finite sums: $\forall j, k \in \mathbb{N}, j \neq k$,
$$\bigoplus_{i \in \emptyset} x_i = \mathbb{0}, \ \bigoplus_{i \in \{j\}} x_i = x_j, \ \bigoplus_{i \in \{j,k\}} x_i = x_j \oplus x_k,$$
ii. associativity and commutativity: for all partition $(I_j)_{j \in J}$ of I,
$$\bigoplus_{j \in J} \bigoplus_{i \in I_j} x_i = \bigoplus_{i \in I} x_i,$$
iii. distributivity of products over infinite sums: for all $I \subseteq \mathbb{N}$, $\forall x, y \in \mathbb{S}$,
$$\bigoplus_{i \in I} (x \otimes y_i) = x \otimes \bigoplus_{i \in I} y_i, \text{ and } \bigoplus_{i \in I} (x_i \otimes y) = (\bigoplus_{i \in I} x_i) \otimes y.$$

Label Theories. The functions labelling the transitions of sw-automata and transducers generalize the Boolean algebras of [8]. We consider *alphabets*, which are non-empty countable sets of symbols denoted by Σ, Δ... and write Σ^* for the set of finite sequences (*words*) over Σ, ε for the empty word, $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, and uv for the concatenation of $u, v \in \Sigma^*$.

Given a semiring $\langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1} \rangle$, a *label theory* $\bar{\Phi}$ over $\mathbb{S}$ is an indexed family of sets Φ_Σ, resp. $\Phi_{\Sigma,\Delta}$, containing recursively enumerable functions of type $\Sigma \to \mathbb{S}$, resp. $\Sigma \times \Delta \to \mathbb{S}$, and such that if $\Phi_{\Sigma,\Delta} \in \bar{\Phi}$, then $\Phi_\Sigma \in \bar{\Phi}$ and $\Phi_\Delta \in \bar{\Phi}$, every $\Phi_\Sigma, \Phi_{\Sigma,\Delta} \in \bar{\Phi}$ contains all the constant functions of $\Sigma \to \mathbb{S}$, resp. $\Sigma \times \Delta \to \mathbb{S}$, for all $\Phi_{\Sigma,\Delta} \in \bar{\Phi}$, $\eta \in \Phi_{\Sigma,\Delta}$, $a \in \Sigma$, $b \in \Delta$, the partial application $x \mapsto \eta(x, b)$ is in Φ_Σ and the partial application $y \mapsto \eta(a, y)$ is in Φ_Δ, and $\bar{\Phi}$ is closed under the following operators, derived from the operations of $\mathbb{S}$:

- For all $\Phi_\Sigma \in \bar{\Phi}$, all $\phi \in \Phi_\Sigma$, and $\alpha \in \mathbb{S}$, $\alpha \otimes \phi : x \mapsto \alpha \otimes \phi(x)$, and $\phi \otimes \alpha : x \mapsto \phi(x) \otimes \alpha$ are in Φ_Σ, and similarly for $\oplus$ and for $\Phi_{\Sigma,\Delta}$.
- For all $\Phi_\Sigma \in \bar{\Phi}$, all $\phi, \phi' \in \Phi_\Sigma$, $\phi \otimes \phi' : x \mapsto \phi(x) \otimes \phi'(x)$ is in Φ_Σ.
- For all $\Phi_{\Sigma,\Delta} \in \bar{\Phi}$, all $\eta, \eta' \in \Phi_{\Sigma,\Delta}$, $\eta \otimes \eta' : x, y \mapsto \eta(x,y) \otimes \eta'(x,y)$ is in $\Phi_{\Sigma,\Delta}$.
- For all $\Phi_\Sigma, \Phi_{\Sigma,\Delta} \in \bar{\Phi}$, all $\phi \in \Phi_\Sigma$ and $\eta \in \Phi_{\Sigma,\Delta}$, $\phi \otimes_1 \eta : x, y \mapsto \phi(x) \otimes \eta(x,y)$ and $\eta \otimes_1 \phi : x, y \mapsto \eta(x,y) \otimes \phi(x)$ are in $\Phi_{\Sigma,\Delta}$.
- For all $\Phi_\Delta, \Phi_{\Sigma,\Delta} \in \bar{\Phi}$, all $\psi \in \Phi_\Delta$ and $\eta \in \Phi_{\Sigma,\Delta}$, $\psi \otimes_2 \eta : x, y \mapsto \psi(y) \otimes \eta(x,y)$ and $\eta \otimes_2 \psi : x, y \mapsto \eta(x,y) \otimes \psi(y)$ are in $\Phi_{\Sigma,\Delta}$.
- Analogous closures hold for $\oplus$.

Example 2. We go back to Example 1. In order to align an input in Σ^* with a music score in Δ^*, we must account for the expressive timing of human performance that results in small time shifts between an input event of Σ and a notational event in Δ. These shifts can be weighted as a distance in $\Phi_{\Sigma,\Delta}$, defined in the tropical min-plus semiring by $\delta(e{:}\,\tau, a{:}\,\tau') = |\tau' - \tau|$ if a corresponds to e (e.g. e is the MIDI key 69 and a is the note A_4), or $\mathbb{0}$ otherwise. $\diamond$

3 SW Visibly Pushdown Automata and Transducers

Let $\langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1}\rangle$ be a commutative and complete semiring and let Σ and Δ be countable alphabets called *input* and *output* respectively, such that Δ is partitioned into three disjoint subsets of symbols Δ_i, Δ_c and Δ_r, called respectively *internal*, *call* and *return* [2]. Let $\bar{\Phi}$ be a label theory over $\mathbb{S}$, consisting of $\Phi_\mathsf{e} = \Phi_\Sigma$, $\Phi_\mathsf{i} = \Phi_{\Delta_\mathsf{i}}$, $\Phi_\mathsf{c} = \Phi_{\Delta_\mathsf{c}}$, $\Phi_\mathsf{r} = \Phi_{\Delta_\mathsf{r}}$, $\Phi_\mathsf{ei} = \Phi_{\Sigma,\Delta_\mathsf{i}}$ and $\Phi_\mathsf{cr} = \Phi_{\Delta_\mathsf{c},\Delta_\mathsf{r}}$.

Definition 1 (swVPT). *A Symbolic Weighted Visibly Pushdown Transducer over Σ, Δ, $\mathbb{S}$, and $\bar{\Phi}$ is a tuple $T = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$, where Q is a finite set of states, P is a finite set of stack symbols, $\mathsf{in} : Q \to \mathbb{S}$ (respectively $\mathsf{out} : Q \to \mathbb{S}$) are functions defining the weight for entering (respectively leaving) a state, and $\bar{\mathsf{w}}$ is a tuplet composed of the transition functions : $\mathsf{w}_{10} : Q\times Q \to \Phi_\mathsf{e}$, $\mathsf{w}_{01} : Q\times Q \to \Phi_\mathsf{i}$, $\mathsf{w}_{11} : Q\times Q \to \Phi_\mathsf{ei}$, $\mathsf{w}_\mathsf{c} : Q\times Q\times P \to \Phi_\mathsf{c}$, $\mathsf{w}_\mathsf{r} : Q\times P\times Q \to \Phi_\mathsf{cr}$, $\mathsf{w}^\mathsf{e}_\mathsf{r} : Q\times Q \to \Phi_\mathsf{r}$.*

For convenience, we extend the above transition functions as follows, for every $q, q' \in Q$, $p \in P$, $e \in \Sigma$, $a \in \Delta_\mathsf{i}$, $c \in \Delta_\mathsf{c}$, $r \in \Delta_\mathsf{r}$, overloading their names:

$$
\begin{aligned}
\mathsf{w}_{10}(q, e, \varepsilon, q') &= \phi(e) && \text{where } \phi = \mathsf{w}_{10}(q, q'),\\
\mathsf{w}_{01}(q, \varepsilon, a, q') &= \phi(a) && \text{where } \phi = \mathsf{w}_{01}(q, q'),\\
\mathsf{w}_{11}(q, e, a, q') &= \eta(e, a) && \text{where } \eta = \mathsf{w}_{11}(q, q'),\\
\mathsf{w}_\mathsf{c}(q, \varepsilon, c, q', p) &= \phi(c) && \text{where } \phi = \mathsf{w}_\mathsf{c}(q, q', p),\\
\mathsf{w}_\mathsf{r}(q, c, p, \varepsilon, r, q') &= \eta(c, r) && \text{where } \eta = \mathsf{w}_\mathsf{r}(q, p, q'),\\
\mathsf{w}^\mathsf{e}_\mathsf{r}(q, \varepsilon, r, q') &= \phi(r) && \text{where } \phi = \mathsf{w}^\mathsf{e}_\mathsf{r}(q, q').
\end{aligned}
$$

The swVPT T computes asynchronously on pairs $\langle s, t\rangle \in \Sigma^* \times \Delta^*$. Intuitively, a transition $\mathsf{w}_{ij}(q, e, a, q')$, with $i, j \in \{0, 1\}$ and $e \in \Sigma \cup \{\varepsilon\}$, $a \in \Delta_\mathsf{i} \cup \{\varepsilon\}$, is interpreted as follows: when reading e and a in the input and output words, it increments the current position in the input word if and only if $i = 1$, and in the output word iff $j = 1$, and changes state from q to q'. When $e = \varepsilon$ (resp. $a = \varepsilon$), the current symbol in the input (resp. output) is not read. These transitions ignore the stack.
A transition of $\mathsf{w}_\mathsf{c}(q, \varepsilon, c, q', p)$ reads the call symbol $c \in \Delta_\mathsf{c}$ in the output word, pushes it to the stack along with $p \in P$, and changes state from q to q'. As for $\mathsf{w}_\mathsf{r}(q, c, p, \varepsilon, r, q')$ and $\mathsf{w}^\mathsf{e}_\mathsf{r}(q, \varepsilon, r, q')$ (used when the stack is empty), they read the return symbol r in the output word and change state from q to q'. Additionally, w_r reads and pops from the stack a pair $\langle c, p\rangle$ and the symbol c is compared to r by the function $\eta = \mathsf{w}_\mathsf{r}(q, p, q') \in \Phi_\mathsf{cr}$.

Formally, the computations of the transducer T are defined with an intermediate function weight_T. A configuration $q[\gamma]$ is composed of a state $q \in Q$ and a stack content $\gamma \in \Gamma^*$, where $\Gamma = \Delta_\mathsf{c} \times P$, and weight_T is a function from $[Q\times \Gamma^*] \times \Sigma^* \times \Delta^* \times [Q\times\Gamma^*]$ into $\mathbb{S}$, whose recursive definition enumerates each of the possible cases for reading $e \in \Sigma$, $a \in \Delta_\mathsf{i}$, $c \in \Delta_\mathsf{c}$, or $r \in \Delta_\mathsf{r}$ (the empty stack is denoted by $\perp$, and the topmost symbol is the last pushed pair):

$$
\mathsf{weight}_T\big(q[\gamma], \varepsilon, \varepsilon, q'[\gamma']\big) = \mathbb{1} \text{ if } q = q', \gamma = \gamma' \text{ and } \mathbb{0} \text{ otherwise} \qquad (1)
$$

$$\begin{aligned}
\mathsf{weight}_T\big(q[\gamma], e\,u, \varepsilon, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{10}(q, e, \varepsilon, q'') \otimes \mathsf{weight}_T\big(q''[\gamma], u, \varepsilon, q'[\gamma']\big)\\
\mathsf{weight}_T\big(q[\gamma], \varepsilon, a\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{01}(q, \varepsilon, a, q'') \otimes \mathsf{weight}_T\big(q''[\gamma], \varepsilon, v, q'[\gamma']\big)\\
\mathsf{weight}_T\big(q[\gamma], e\,u, a\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{10}(q, e, \varepsilon, q'') \otimes \mathsf{weight}_T\big(q''[\gamma], u, a\,v, q'[\gamma']\big)\\
&\oplus \bigoplus_{q''\in Q} \mathsf{w}_{01}(q, \varepsilon, a, q'') \otimes \mathsf{weight}_T\big(q''[\gamma], e\,u, v, q'[\gamma']\big)\\
&\oplus \bigoplus_{q''\in Q} \mathsf{w}_{11}(q, e, a, q'') \otimes \mathsf{weight}_T\big(q''[\gamma], u, v, q'[\gamma']\big)\\
\mathsf{weight}_T\big(q[\gamma], u, c\,v, q'[\gamma']\big) &= \bigoplus_{\substack{q''\in Q\\ p\in P}} \mathsf{w}_{\mathsf{c}}(q, \varepsilon, c, q'', p) \otimes \mathsf{weight}_T\Big(q'' \begin{bmatrix}\langle c, p\rangle\\ \gamma\end{bmatrix}, u, v, q'[\gamma']\Big)\\
\mathsf{weight}_T\Big(q \begin{bmatrix}\langle c, p\rangle\\ \gamma\end{bmatrix}, u, r\,v, q'[\gamma']\Big) &= \bigoplus_{q''\in Q} \mathsf{w}_{\mathsf{r}}\big(q, c, p, \varepsilon, r, q''\big) \otimes \mathsf{weight}_T\big(q''[\gamma], u, v, q'[\gamma']\big)\\
\mathsf{weight}_T\big(q[\bot], u, r\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}^{\mathsf{e}}_{\mathsf{r}}(q, \varepsilon, r, q'') \otimes \mathsf{weight}_T\big(q''[\bot], u, v, q'[\gamma']\big)
\end{aligned}$$

We recall that, by convention, an empty sum with $\bigoplus$ is equal to $\mathbb{0}$.
The weight associated by T to an input/output pair $\langle s, t\rangle \in \Sigma^* \times \Delta^*$ is defined according to empty stack semantics:

$$T(s, t) = \bigoplus_{q, q'\in Q} \mathsf{in}(q) \otimes \mathsf{weight}_T\big(q[\bot], s, t, q'[\bot]\big) \otimes \mathsf{out}(q') \tag{2}$$

Since $\mathbb{0}$ is absorbing for $\otimes$, and neutral for $\oplus$ in $\mathbb{S}$, if a transition's weight is equal to $\mathbb{0}$, then the entire term is $\mathbb{0}$, meaning the transition is impossible. This is analogous to the case of a transition's guard not satisfied in symbolic models [8].

Symbolic Weighted Visibly Pushdown Automata. swVPA are the particular case of swVPT that do not read in the input word, i.e. where all w_{10} and w_{11} are constant functions equal to $\mathbb{0}$, or, equivalently, $\Sigma = \emptyset$ (see Appendix C for details). They are a weighted extension of sVPA [7], from Boolean semirings to arbitrary semiring domains. A relationship between swVPA and sw-Tree Automata is presented in Appendix F.

Example 3. We consider a swVPA A over Δ^*, with $P = Q$, computing a value of *notational complexity* for a given score. In a sequence $O \in \Delta^*$ like in Example 1, Δ_{i} contains timed notes and continuations, and Δ_{c} and Δ_{r} contain respectively opening and closing parentheses. To a call symbol $\lceil_n\!: \ell$, for some duration value ℓ, let us associate a transition for the division of ℓ by n: $\mathsf{w}_{\mathsf{c}}\big(q_\ell, \varepsilon, \lceil_n\!: \ell, q_{\frac{\ell}{n}}, q'\big) = \alpha_n \in \mathbb{S}$. And to the matching return symbol $\rceil_n\!: \ell$, we associate a transition of weight $\mathbb{1}$: $\mathsf{w}_{\mathsf{r}}\big(q_{\frac{\ell}{n}}, \lceil_n\!: \ell, q', \varepsilon, \rceil_n\!: \ell, q'\big) = \mathbb{1}$, which jumps to the state q' stored in the stack. The choice of weight values for the call transitions can express some preferences in term of the expected output notation: if we want to prioritize pairs over triplets, in the Tropical min-plus semiring, then we would let $\alpha_2 < \alpha_3$. It is able to compute on several representations of a piece of music, estimating for each one a weight value depending on the preferences that we set. The algorithm of Theorem 4 then allows to select the best weighted representation. $\diamond$

Symbolic Weighted Transducers. swT are the particular case of swVPT that do not use the stack during their computations, because all $\mathsf{w_c}$, $\mathsf{w_r}$ and $\mathsf{w_r^e}$ are constant functions equal to $\mathbb{0}$, or, more simply, because $\Delta_\mathsf{c} = \Delta_\mathsf{r} = \emptyset$ (see App C).

The four first lines in expression (1) can be seen as a stateful definition of an edit-distance between a word $s \in \Sigma^*$ and a word $t \in {\Delta_\mathsf{i}}^*$, see also [17]. Intuitively, in this vision, $\mathsf{w}_{10}(q, e, \varepsilon, q')$ is the cost of the deletion of the symbol $e \in \Sigma$ in s, $\mathsf{w}_{01}(q, \varepsilon, a, q')$ is the cost of the insertion of $a \in \Delta_\mathsf{i}$ in t, and $\mathsf{w}_{11}(q, e, a, q')$ is the cost of the substitution of $e \in \Sigma$ by $a \in \Delta_\mathsf{i}$. Following (2), the cost of a sequence of such operations transforming s into t is the product in terms of $\otimes$ of the individual costs of the operations involved, and the distance between s and t is the sum in terms of $\oplus$ of all possible products.

Example 4. We propose a swT over Σ and Δ_i that computes the distance between an input $I \in \Sigma^*$ and an output $O \in {\Delta_\mathsf{i}}^*$ like in Ex. 1 (for δ, see Ex. 2):

$$\mathsf{w}_{11}(q_0, e{:}\,\tau, a{:}\,\tau', q_0) \text{ and } \mathsf{w}_{11}(q_1, e{:}\,\tau, a{:}\,\tau', q_0) = \delta(e{:}\,\tau, a{:}\,\tau') \quad \text{if } a \neq -$$
$$\mathsf{w}_{01}(q_0, \varepsilon, -{:}\,\tau', q_0) = \mathbb{1} \quad \mathsf{w}_{01}(q_1, \varepsilon, -{:}\,\tau', q_0) = \mathbb{1} \quad \mathsf{w}_{10}(q_0, e{:}\,\tau, \varepsilon, q_1) = \alpha$$

The continuation symbol $'-'$ (e.g. in *ties* ♩‿♪, or *dots* ♩.) is skipped with no cost (w_{01}). We also want to consider performing errors, by switching to an error state q_1. Reading an extra event e is handled by w_{10} that switches to q_1, with a fixed $\alpha \in \mathbb{S}$, then w_{11} and w_{01} can switch back to q_0. Finally, we let q_0 be the initial and final state, with $\mathsf{in}(q_0) = \mathsf{out}(q_0) = \mathbb{1}$, and $\mathsf{in}(q_1) = \mathsf{out}(q_1) = \mathbb{0}$. $\diamond$

Symbolic Weighted Automata. swA are particular cases of swT omitting the output symbols, or equivalently, swVPA without markups ($\Delta_\mathsf{c} = \Delta_\mathsf{r} = \emptyset$).

4 Symbolic Weighted Parsing

Parsing is the problem of structuring a linear representation (a finite word) according to a language model [11]. We shall consider in this section the problem of parsing over an infinite alphabet. Let $\mathbb{S}$ Σ, Δ, and $\bar{\Phi}$ be like in Section 3. We assume to be given the following input:

- a swT T over Σ, Δ_i, $\mathbb{S}$, and $\bar{\Phi}$, defining a measure $T : \Sigma^* \times {\Delta_\mathsf{i}}^* \to \mathbb{S}$,
- a swVPA A over Δ, $\mathbb{S}$, and $\bar{\Phi}$, defining a measure $A : \Delta^* \to \mathbb{S}$,
- an (unstructured) input word $s \in \Sigma^*$.

For every $u \in \Sigma^*$ and $t \in \Delta^*$, let $d_T(u,t) = T\big(u, t|_{\Delta_\mathsf{i}}\big)$, where $t|_{\Delta_\mathsf{i}} \in {\Delta_\mathsf{i}}^*$ is the projection of t onto Δ_i, obtained from t by removing all symbols in $\Delta \setminus \Delta_\mathsf{i}$. Given the above input, *symbolic weighted parsing* aims at of finding a (structured) nested word $t \in \Delta^*$ that minimizes $d(s,t) \otimes A(t)$ wrt $\leq_\oplus$, i.e. such that:

$$d_T(s,t) \otimes A(t) = \bigoplus_{v \in \Delta^*} d_T(s,v) \otimes A(u) \tag{3}$$

In the terminology of [17], sw-parsing is the problem of computing the distance (3) between the input word s and the weighted language over the output alphabet defined by A, and returning a witness t.

Example 5. In the running example, the input is as follows: The swT T evaluates a "fitness measure": a temporal distance between a performance and a nested-word representation of a music score (Example 4). The swVPA A expresses a weight related to the complexity of music notation (Example 3). The input word is I of Example 1. The notation , will be favored over  when the weight assigned to the call $\lceil_2$ is less than the difference of weight between the appogiatura '$\mathsf{c}\sharp_5$' and the standard note $\mathsf{c}\sharp_5$. The sw-parsing framework, applied to music transcription, finds an optimal solution considering both the fitness of the output to the input, and its notational complexity. $\diamond$

Nested words in Δ^* can represent linearizations of labeled trees, and to any Weighted Regular Tree Grammar (wRTG), we can associate in polynomial time a swVPA computing the same weight (see Appendix F). Therefore, instead of a swVPA in input, we may be given a wRTG, or a weighted CFG (wCFG), for a definition closer to conventional parsing. The sw-parsing problem hence generalizes the problem of searching for the best derivation tree of a wCFG G that yields a given input word w, with an infinite input alphabet instead of a finite one and transducer-defined distances instead of equality. It is however uncomparable to the related problems of *semiring parsing* [10], and *weighted parsing* [18].

In Section 5, we present results on swVPT and subclasses (automata construction and best-search algorithm) that can be applied for solving sw-parsing.

Theorem 1. *The problem of Symbolic Weighted Parsing can be solved in PTIME in the size of the input swT T, swVPA A and input word s, and the computation time of the functions and operators of the label theory.*

Proof. We follow an approach of *parsing as intersection* [11, ch. 13]. First, we associate to T and A a swVPT called $(T \otimes A)$, computing the product of the respective weights for the two models (Theorem 2): i.e. $(T \otimes A)(u,t) = d_T(u,t) \otimes A(t)$. Then, we construct a swVPA computing, for $t \in \Delta^*$, $(T \otimes A)(s,t) = d_T(s,t) \otimes A(t)$ (Theorem 3). Finally, with the algorithm of Theorem 4, we find a *best* $t \in \Delta^*$ minimizing the latter value wrt $\leq_\oplus$, i.e. a solution of sw-parsing. $\square$

5 Properties and Best-Search Algorithm

In the following results, we assume that the functions of a label theory $\bar{\Phi}$ are given in a finite representation (e.g. Turing machine) in the definitions of swVPT, and provide complexity bounds parameterized by the semiring operators and the operators of Section 2 over the functions of $\bar{\Phi}$ (the latter might be represented symbolically by structures like Algebraic Decision Diagrams [4]).
Similarly to VPA [2] and sVPA [7], the class of swVPT is closed under the binary operators of the underlying semiring.

Proposition 1. *Let T_1, T_2 be two swVPT over the same Σ, Δ, commutative $\mathbb{S}$ and $\bar{\Phi}$. There exist two effectively constructible swVPT $T_1 \oplus T_2$ and $T_1 \otimes T_2$, such that for every $s \in \Sigma^*$ and $t \in \Delta^*$, $(T_1 \oplus T_2)(s,t) = T_1(s,t) \oplus T_2(s,t)$ and $(T_1 \otimes T_2)(s,t) = T_1(s,t) \otimes T_2(s,t)$.*

Proof. Classical Cartesian product construction, similar to the case of the Boolean semiring [7], see Appendix B for details. ☐

The following result shows how to compose, in a single swVPT, the two measures as input of sw-parsing: the swT computing input-output distance and the swVPA expressing the weight of parse trees' linearization.

Theorem 2. *Given a swT T over Σ, Δ_i, commutative $\mathbb{S}$, and $\bar{\Phi}$, and a swVPA A over Δ, $\mathbb{S}$, $\bar{\Phi}$, one can construct in PTIME a swVPT $T \otimes A$, over Σ, Δ, $\mathbb{S}$, $\bar{\Phi}$, such that $\forall s \in \Sigma^*$, $t \in \Delta^*$, $(T \otimes A)(s,t) = T(s, t|_{\Delta_\mathsf{i}}) \otimes A(t)$.*

Proof. (sketch, see Appendix C for details). The state set of $T \otimes A$ is the Cartesian product of the state sets of T and A, and every transition of $T \otimes A$ is either a transition of T or a transition of A of the same kind (in these cases the state of the other machine remains the same), or a product of two transitions w_{11} of T and A. ☐

The next result is the construction, as a swVPA, for the partial application of a swVPT, fixing an input word s as its first argument.

Theorem 3. *Given a swVPT T over Σ, Δ, commutative, complete and idempotent $\mathbb{S}$, and $\bar{\Phi}$, and given $s \in \Sigma^*$, there exists an effectively constructible swVPA $T(s)$ over Δ, $\mathbb{S}$, and $\bar{\Phi}$, such that for every $t \in \Delta^*$, $T(s)(t) = T(s,t)$.*

Proof. (sketch, see Appendix D). We construct an automaton that simulates, while reading an output word $t \in \Delta^*$, the synchronized computation of T on s and t. The main difficulty comes from the transitions of T of the form w_{10}, which read in input s and ignore the output t. Since the automaton $A(T)$ only reads the output word t, we add to $A(T)$ a corresponding ε-transition, and show how to remove the ε-transitions from a swVPA while preserving its language. ☐

We present now a procedure for searching a word of minimal weight for a swVPA A, a variant of reachability problems in pushdown automata [5].

First of all, for a complete semiring $\mathbb{S}$, we consider the following operators on the functions of a label theory $\bar{\Phi}$:

$$\bigoplus_{\Sigma} : \Phi_{\Sigma} \to \mathbb{S},\ \phi \mapsto \bigoplus_{a \in \Sigma} \phi(a) \quad \bigoplus^1_{\Sigma} : \Phi_{\Sigma,\Delta} \to \Phi_{\Delta},\ \eta \mapsto \big(y \mapsto \bigoplus_{a \in \Sigma} \eta(a, y)\big)$$
$$\bigoplus^2_{\Delta} : \Phi_{\Sigma,\Delta} \to \Phi_{\Sigma},\ \eta \mapsto \big(x \mapsto \bigoplus_{b \in \Delta} \eta(x, b)\big)$$

Intuitively, $\bigoplus_{\Sigma}$ returns the global minimum, wrt $\leq_{\oplus}$, of a function ϕ of Φ_{Σ}, and $\bigoplus^1_{\Sigma}$, $\bigoplus^2_{\Delta}$ return partial minimums of a function η of $\Phi_{\Sigma,\Delta}$. A label theory is called *effective* when the three above operators applied on its functions are recursively enumerable, and there exists a function returning a witness symbol that reaches the minimum. In the complexity bounds, we assume a constant time evaluation for these operators. Effectiveness of label theories is a strong restriction, although realistic for the case study presented in this paper. It is satisfied e.g. by functions with a codomain $\{\mathbb{0}, \alpha\}$, with $\alpha <_{\oplus} \mathbb{0}$, generalizing the boolean guards of [7,8] to *filters* returning null or constant weight values.

Theorem 4. *For a* swVPA *A over Δ, $\mathbb{S}$ commutative, complete, bounded and total, and $\bar{\Phi}$ effective, one can construct in PTIME a word $t \in \Delta^*$ such that $A(t)$ is minimal wrt the natural ordering $\leq_{\oplus}$ for $\mathbb{S}$.*

Proof. Let $A = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$. For every $q, q' \in Q$, let $b_{\perp}(q, q')$ be the minimum, wrt $\leq_{\oplus}$, of the function $\beta_{q,q'} : t \mapsto \mathsf{weight}_A(q[\perp], \varepsilon, t, q'[\perp])$. By definition of $\leq_{\oplus}$, and since $\mathbb{S}$ is complete and total, it holds that: $b_{\perp}(q, q') = \bigoplus_{t \in \Delta^*} \mathsf{weight}_A\big(q[\perp], \varepsilon, t, q'[\perp]\big)$ (see (1) for the definition of weight_A). Following (2), and the algebraic properties of $\otimes$ and $\oplus$, the minimum of $A(t)$ wrt $\leq_{\oplus}$ is:

$$\bigoplus_{t \in \Delta^*} A(t) = \bigoplus_{t \in \Delta^*} \bigoplus_{q,q' \in Q} \mathsf{in}(q) \otimes \beta_{q,q'}(t) \otimes \mathsf{out}(q') = \bigoplus_{q,q' \in Q} \mathsf{in}(q) \otimes b_{\perp}(q, q') \otimes \mathsf{out}(q') \tag{4}$$

Hence, in order to prove Theorem 4, it is sufficient to show that for all $q, q' \in Q$, we can compute $b_{\perp}(q, q')$ in PTIME. We proceed by searching for a best weighted derivation in a $\mathbb{S}$-labeled hypergraph $\mathcal{G}_A$ associated to the swVPA A. It has a set of vertices $V_A = (Q \times \{\perp, \top\} \times Q)$, where $\top$ is a new symbol representing a non-empty stack, a set of hyperedges $E_A = (V_A \times V_A) \cup (V_A \times V_A \times V_A)$, and an hyperedge labelling function $\eta_A : E_A \to \mathbb{S}$ defined as follows, for $q_0, q'_0, q_1, q_2, q_3 \in Q$, ($\mathsf{w_i}$ is another name for w_{01}, like in Appendix C, Definition 3):

$$\begin{aligned}
&\langle q_0, \perp, q_1\rangle, \langle q'_0, \gamma, q_2\rangle \mapsto \mathbb{0} \quad \text{if } \gamma = \top \text{ or } (\gamma = \perp \text{ and } q'_0 \neq q_0)\\
&\langle q_0, \perp, q_1\rangle, \langle q_0, \perp, q_2\rangle \mapsto \textstyle\bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q_1, q_2) \oplus \bigoplus_{\Delta_\mathsf{r}} \mathsf{w^e_r}(q_1, q_2)\\
&\langle q_1, \top, q_2\rangle, \langle q_0, \perp, q_3\rangle \mapsto \bigoplus_{p \in P} \textstyle\bigoplus_{\Delta_\mathsf{c}} \big[\mathsf{w_c}(q_0, q_1, p) \otimes \bigoplus^2_{\Delta_\mathsf{r}} \mathsf{w_r}(q_2, p, q_3)\big]\\
&\langle q_1, \top, q_2\rangle, \langle q_0, \top, q_3\rangle \mapsto \Big[\bigoplus_{q_1 = q_0} \textstyle\bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q_2, q_3)\Big] \oplus\\
&\qquad\qquad\Big[\bigoplus_{p \in P} \textstyle\bigoplus_{\Delta_\mathsf{c}} \big[\mathsf{w_c}(q_0, q_1, p) \otimes_2 \bigoplus^2_{\Delta_\mathsf{r}} \mathsf{w_r}(q_2, p, q_3)\big]\Big]\\
&\langle q_0, \gamma_1, q_1\rangle, \langle q_1, \gamma_2, q_2\rangle, \langle q_0, \gamma, q_2\rangle \mapsto \mathbb{1} \quad \text{if } \gamma_1 = \gamma_2 = \gamma \text{ or } \mathbb{0} \text{ otherwise.}
\end{aligned}$$

Intuitively, a vertex $v = \langle q, \bot, q' \rangle$ (resp. $v = \langle q, \top, q' \rangle$) of $\mathcal{G}_A$ represents computations of $\mathcal{A}$ starting in state q with an empty stack (resp. non-empty stack γ), and ending in state q' with an empty stack (resp. the same non-empty stack γ). The best weight of such computations is the best cumulated weight of hyperedges along a derivation to v. More precisely, a *derivation* of $\mathcal{G}_A$ is a V_A-labeled binary tree of the form, v, $v(\theta_1)$ or $v(\theta_1, \theta_2)$, where θ_1 and θ_2 are sub-derivations, and its weight is defined by (for $i = 1, 2$, the root of θ_i, is labeled with $v_i \in V_A$):

- $\mathsf{weight}(\langle q, \bot, q \rangle) = \mathsf{weight}(\langle q, \top, q \rangle) = \mathbb{1}$,
- $\mathsf{weight}(\langle q, \bot, q' \rangle) = \mathsf{weight}(\langle q, \top, q' \rangle) = \mathbb{0}$ if $q \neq q'$,
- $\mathsf{weight}\big(v(\theta_1)\big) = \mathsf{weight}(\theta_1) \otimes \eta_A(v_1, v)$
- $\mathsf{weight}\big(v(\theta_1, \theta_2)\big) = \mathsf{weight}(\theta_1) \otimes \mathsf{weight}(\theta_2) \otimes \eta_A(v_1, v_2, v)$.

With $\mathcal{D}(\mathcal{G}_A, v)$ denoting the set of derivations of $\mathcal{G}_A$ with root labeled with $v \in V_A$, it holds that (see Appendix E for the proof of the following Lemma 4):

Lemma 4. *For all $q, q' \in Q$, $b_\bot(q, q') = \displaystyle\bigoplus_{\theta \in \mathcal{D}(\mathcal{G}_A, \langle q, \bot, q' \rangle)} \mathsf{weight}(\theta)$.*

Therefore, computing $b_\bot(q, q')$ reduces to the search for a smallest weighted derivation of $\mathcal{G}_A$ (wrt $\leq_\oplus$) rooted with $\langle q, \bot, q' \rangle$, a problem solvable in PTIME [13], because $\mathbb{S}$ is monotonic wrt $\leq_\oplus$ and superior (Lemma 2). Therefore, by (4), the minimum of $t \mapsto A(t)$, wrt $\leq_\oplus$, can be computed in PTIME.
Moreover, a witness $t \in \Delta^*$ for this minimum can be associated to the appropriate best derivation, with no additional cost. For details on the extraction of this witness, see Appendix E, the proof of Lemma 4, and Lemma 6. ☐

Conclusion

We presented closure properties and one decision algorithm for three classes of Symbolic Weighted language models: swVPT, swT and swVPA, and applied these results to the problem of parsing with infinitely many input symbols (typically timed events). In our approach to parsing, words are compared by computing a distance between them, defined by a given sw-transducer, which allows to consider finer word relationships than strict equality.

The application to automated music transcription suggested in a toy example has been implemented in a C++ library [1], following the principles of the present sw-parsing framework, although some differences; e.g. the automata constructions are performed on-the-fly during best-search for efficiency reasons. One advantage of this swVPA approach is the *global view* provided by the stack during transcription, as opposed to other HMM-based approaches [22].

This work can be extended in several directions. The best-search algorithm for swVPA could be generalized from 1-best to n-best [14], and to k-*closed* semirings [16] (instead of *bounded*, which corresponds to 0-*closed*). One could also study the generalization of the best-search algorithm of Theorem 4 to the computation of the best possible output of a swVPT for a given input, or even to the more general models of [12].

Finally, the best-search algorithm presented here works offline, whereas an on-the-fly approach coupling automata construction and best-search would be interesting e.g. for online XML validation or filtering, or program monitoring [7].

Acknowledgments.

The authors would like to thank the reviewers at CIAA for their useful remarks.

References

1. library qparse for music transcription. `https://qparse.gitlabpages.inria.fr`
2. Alur, R., Madhusudan, P.: Adding Nesting Structure to Words. Journal of the ACM **56**(3), 1–43 (2009)
3. Alur, R., Mamouras, K., Stanford, C.: Automata-Based Stream Processing. In: 44th ICALP. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2017)
4. Bahar, R.I.e.a.: Algebraic decision diagrams and their applications. Formal methods in system design **10**(2-3), 171–206 (1997)
5. Bouajjani, A., Esparza, J., Maler, O.: Reachability analysis of pushdown automata: Application to model-checking. In: CONCUR, LNCS, vol. 1243. Springer (1997)
6. Caralp, M., Reynier, P.A., Talbot, J.M.: Visibly Pushdown Automata with Multiplicities: finiteness and k-boundedness. In: DLT. Springer (2012)
7. D'Antoni, L., Alur, R.: Symbolic Visibly Pushdown Automata. In: International Conference on Computer Aided Verification. pp. 209–225. Springer (2014)
8. D'Antoni, L., Veanes, M.: The power of symbolic automata and transducers. In: International Conference on Computer Aided Verification. pp. 47–67. Springer (2017)
9. Droste, M., Kuich, W., Vogler, H.: Handbook of Weighted Automata. Springer Science & Business Media (2009)
10. Goodman, J.: Semiring Parsing. Computational Linguistics **25**(4), 573–606 (1999)
11. Grune, D., Jacobs, C.J.: Parsing Techniques. No. 2nd edition in Monographs in Computer Science, Springer (2008)
12. Herrmann, L., Vogler, H.: Weighted symbolic automata with data storage. In: DLT. Springer (2016)
13. Huang, L.: Advanced dynamic programming in semiring and hypergraph frameworks. In: COLING (2008)
14. Huang, L., Chiang, D.: Better k-best parsing. In: Proceedings of the 9th International Workshop on Parsing Technology. ACL (2005)
15. Ma, M., Du, D., Liu, Y., Wang, Y., Li, Y.: Efficient Adversarial Sequence Generation for RNN with Symbolic Weighted Finite Automata. In: Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI). vol. 3087 (2022)
16. Mohri, M.: Semiring frameworks and algorithms for shortest-distance problems. Journal of Automata, Languages and Combinatorics **7**(3), 321–350 (2002)
17. Mohri, M.: Edit-distance of weighted automata: General definitions and algorithms. International Journal of Foundations of Computer Science **14**(06), 957–982 (2003)
18. Mörbitz, R., Vogler, H.: Weighted parsing for grammar-based language models. In: 14th Int. Conf. on Finite-State Methods and Natural Language Proc. ACL (2019)
19. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, vol. 1 word, language, grammar. Springer-Verlag New York, Inc. (1997)
20. Segoufin, L.: Automata and logics for words and trees over an infinite alphabet. In: Computer Science Logic. LNCS, vol. 4207. Springer (2006)

21. Selfridge-Field, E. (ed.): Beyond MIDI: the handbook of musical codes. MIT press (1997), `http://beyondmidi.ccarh.org/beyondmidi-600dpi.pdf`
22. Shibata, K., Nakamura, E., Yoshii, K.: Non-local musical statistics as guides for audio-to-score piano transcription. Information Sciences **566**, 262–280 (2021)
23. Waga, M.: Online quantitative timed pattern matching with semiring-valued weighted automata. In: Int. Conf. on Formal Modeling and Analysis of Timed Systems. Springer (2019)
24. Yust, J.: Organized Time. Oxford University Press (2018)

Appendices

A Lemmata on Semirings

Lemma 2 (Superiority, Boundedness). *Let $\langle \mathbb{S}, \oplus, \mathbb{0}, \otimes, \mathbb{1}\rangle$ be an idempotent semiring. The two following statements are equivalent: (i) for all $x, y \in \mathbb{S}$, $x \leq_\oplus x \otimes y$ and $y \leq_\oplus x \otimes y$ (ii) for all $x \in \mathbb{S}$, $\mathbb{1} \oplus x = \mathbb{1}$.*

Proof. $(ii) \Rightarrow (i)$: $x \oplus (x \otimes y) = x \otimes (\mathbb{1} \oplus y) = x$, by distributivity of $\otimes$ over $\oplus$. Hence $x \leq_\oplus x \otimes y$. Similarly, $y \oplus (x \otimes y) = (\mathbb{1} \oplus x) \otimes y = y$, hence $y \leq_\oplus x \otimes y$. $(i) \Rightarrow (ii)$: by the second inequality of (i), with $y = \mathbb{1}$, $\mathbb{1} \leq_\oplus x \otimes \mathbb{1} = x$, i.e., by definition of $\leq_\oplus$, $\mathbb{1} \oplus x = \mathbb{1}$. □

Lemma 3 ([16], Lemma 3). *Every bounded semiring is idempotent.*

Proof. By boundedness, $\mathbb{1} \oplus \mathbb{1} = \mathbb{1}$, and idempotency follows by multiplying both sides by x and distributing. □

B Proof of Proposition 1

Proposition 1. *Let T_1, T_2 be two swVPT over the same Σ, Δ, commutative $\mathbb{S}$ and $\bar{\Phi}$. There exist two effectively constructible swVPT $T_1 \oplus T_2$ and $T_1 \otimes T_2$, such that for every $s \in \Sigma^*$ and $t \in \Delta^*$, $(T_1 \oplus T_2)(s,t) = T_1(s,t) \oplus T_2(s,t)$ and $(T_1 \otimes T_2)(s,t) = T_1(s,t) \otimes T_2(s,t)$.*

Proof. We prove the closure under $\otimes$ (the case of $\oplus$ is similar).
Let $T_1 = \langle Q_1, P_1, \mathsf{in}_1, \bar{\mathsf{w}}_1, \mathsf{out}_1\rangle$ and $T_2 = \langle Q_2, P_2, \mathsf{in}_2, \bar{\mathsf{w}}_2, \mathsf{out}_2\rangle$. We can build a swVPT $T_1 \otimes T_2$ by a classical product construction. We define $T_1 \otimes T_2 = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$, whose set of states $Q = Q_1 \times Q_2$ is the Cartesian product of the respective sets of states of T_1 and T_2, and auxiliary set of stack symbols $P = P_1 \times P_2$ is the product of their sets of stack symbols. The state entering and leaving functions in, out and the tuplet of transition functions $\bar{\mathsf{w}}$ are defined using the label-theory operators of Section 2 as follows, for all $\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle \in Q$ and $\langle p_1, p_2\rangle, \langle p_1', p_2'\rangle \in P$:

$$
\begin{aligned}
\mathsf{in}\big(\langle q_1, q_2\rangle\big) &= \mathsf{in}_1(q_1) \otimes \mathsf{in}_2(q_2)\\
\mathsf{out}\big(\langle q_1, q_2\rangle\big) &= \mathsf{out}_1(q_1) \otimes \mathsf{out}_2(q_2)\\
\mathsf{w}_{10}\big(\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle\big) &= \mathsf{w}_{10,1}(q_1, q_1') \otimes \mathsf{w}_{10,2}(q_2, q_2')\\
\mathsf{w}_{01}\big(\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle\big) &= \mathsf{w}_{01,1}(q_1, q_1') \otimes \mathsf{w}_{01,2}(q_2, q_2')\\
\mathsf{w}_{11}\big(\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle\big) &= \mathsf{w}_{11,1}(q_1, q_1') \otimes \mathsf{w}_{11,2}(q_2, q_2')\\
\mathsf{w}_{\mathsf{c}}\big(\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle, \langle p_1, p_2\rangle\big) &= \mathsf{w}_{\mathsf{c},1}(q_1, q_1', p_1) \otimes \mathsf{w}_{\mathsf{c},2}(q_2, q_2', p_2)\\
\mathsf{w}_{\mathsf{r}}\big(\langle q_1, q_2\rangle, \langle p_1, p_2\rangle, \langle q_1', q_2'\rangle\big) &= \mathsf{w}_{\mathsf{r},1}(q_1, p_1, q_1') \otimes \mathsf{w}_{\mathsf{r},2}(q_2, p_2, q_2')\\
\mathsf{w}_{\mathsf{r}}^{\mathsf{e}}\big(\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle\big) &= \mathsf{w}_{\mathsf{r},1}^{\mathsf{e}}(q_1, q_1') \otimes \mathsf{w}_{\mathsf{r},2}^{\mathsf{e}}(q_2, q_2')
\end{aligned}
$$

With these functions, T simulates the synchronized behaviour of T_1 and T_2. □

C Proof of Theorem 2

Before giving the details of the construction for the proof of Theorem 2, let us first state explicitly the definitions of the classes swT and swVPA.

Definition 2 (swT). *A* Symbolic Weighted Transducer *over Σ, Δ_{i}, $\mathbb{S}$, and $\bar{\Phi}$ is a tuple $T = \langle Q, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$, where Q is a finite set of states, $\mathsf{in} : Q \to \mathbb{S}$ (respectively $\mathsf{out} : Q \to \mathbb{S}$) are functions defining the weight for entering (respectively leaving) a state, and $\bar{\mathsf{w}}$ is composed of the transition functions : $\mathsf{w}_{10} : Q \times Q \to \Phi_{\mathsf{e}}$, $\mathsf{w}_{01} : Q \times Q \to \Phi_{\mathsf{i}}$, $\mathsf{w}_{11} : Q \times Q \to \Phi_{\mathsf{ei}}$.*

We use the same extended notation for the transition functions w_{10}, w_{01}, w_{11} as in Section 3, Definition 1 and follow the definition of the computed weight in (1) (4 first equations) and (2).

Definition 3 (swVPA). *A* Symbolic Weighted Visibly Pushdown Automaton *over $\Delta = \Delta_{\mathsf{i}} \uplus \Delta_{\mathsf{c}} \uplus \Delta_{\mathsf{r}}$, $\mathbb{S}$ and $\bar{\Phi}$ is a tuple $A = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$, where Q is a finite set of states, P is a finite set of stack symbols, $\mathsf{in} : Q \to \mathbb{S}$ (respectively $\mathsf{out} : Q \to \mathbb{S}$) are functions defining the weight for entering (respectively leaving) a state, and $\bar{\mathsf{w}}$ is a tuplet composed of the transition functions : $\mathsf{w}_{\mathsf{i}} : Q \times Q \to \Phi_{\mathsf{i}}$, $\mathsf{w}_{\mathsf{c}} : Q \times Q \times P \to \Phi_{\mathsf{c}}$, $\mathsf{w}_{\mathsf{r}} : Q \times P \times Q \to \Phi_{\mathsf{cr}}$, $\mathsf{w}_{\mathsf{r}}^{\mathsf{e}} : Q \times Q \to \Phi_{\mathsf{r}}$.*

The transition function w_{i} is just a new name for w_{01} of Definition 1, used in the case of swVPA. In the extended notation for the transition functions after Definition 1, and the definition of the computed weight in (1) and (2), there is no input symbol of Σ to read for swVPA (the corresponding argument is always ε in this case). Hence for the sake of simplicity, let us restate explicitly the equations without symbols of Σ, as follows. For the transition functions in extended notation, we have with $q, q' \in Q$ and $a \in \Delta_{\mathsf{i}}$, $c \in \Delta_{\mathsf{c}}$, $r \in \Delta_{\mathsf{r}}$:

$$
\begin{aligned}
\mathsf{w}_{\mathsf{i}}(q, a, q') &= \phi(a) \quad \text{where } \phi = \mathsf{w}_{\mathsf{i}}(q, q'),\\
\mathsf{w}_{\mathsf{c}}(q, c, q', p) &= \phi(c) \quad \text{where } \phi = \mathsf{w}_{\mathsf{c}}(q, q', p),\\
\mathsf{w}_{\mathsf{r}}(q, c, p, r, q') &= \eta(c, r) \text{ where } \eta = \mathsf{w}_{\mathsf{r}}(q, p, q'),\\
\mathsf{w}_{\mathsf{r}}^{\mathsf{e}}(q, r, q') &= \phi(r) \quad \text{where } \phi = \mathsf{w}_{\mathsf{r}}^{\mathsf{e}}(q, q').
\end{aligned}
$$

And for the weight function, with $v \in \Delta^*$:

$$
\begin{aligned}
\mathsf{weight}_A\big(q[\gamma], \varepsilon, q'[\gamma']\big) &= \mathbb{1} \text{ if } q = q', \gamma = \gamma' \text{ and } \mathbb{0} \text{ otherwise} \qquad (5)\\
\mathsf{weight}_A\big(q[\gamma], a\, v, q[\gamma']\big) &= \bigoplus_{q'' \in Q} \mathsf{w}_{\mathsf{i}}(q, a, q'') \otimes \mathsf{weight}_A\big(q''[\gamma], v, q[\gamma']\big)\\
\mathsf{weight}_A\big(q[\gamma], c\, v, q'[\gamma']\big) &= \bigoplus_{\substack{q'' \in Q\\ p \in P}} \mathsf{w}_{\mathsf{c}}(q, c, q'', p) \otimes \mathsf{weight}_A\Big(q'' \begin{bmatrix} \langle c, p\rangle \\ \gamma \end{bmatrix}, v, q'[\gamma']\Big)\\
\mathsf{weight}_A\Big(q \begin{bmatrix} \langle c, p\rangle \\ \gamma \end{bmatrix}, r\, v, q'[\gamma']\Big) &= \bigoplus_{q'' \in Q} \mathsf{w}_{\mathsf{r}}\big(q, c, p, r, q''\big) \otimes \mathsf{weight}_A\big(q''[\gamma], v, q'[\gamma']\big)
\end{aligned}
$$

$$\mathsf{weight}_A\big(q[\bot], r\,v, q'[\gamma']\big) = \bigoplus_{q''\in Q} \mathsf{w}^{\mathsf{e}}_{\mathsf{r}}(q, r, q'') \otimes \mathsf{weight}_A\big(q''[\bot], v, q'[\gamma']\big)$$

and, for $t \in \Delta^*$:

$$A(t) = \bigoplus_{q,q'\in Q} \mathsf{in}(q) \otimes \mathsf{weight}_A\big(q[\bot], t, q'[\bot]\big) \otimes \mathsf{out}(q') \tag{6}$$

We recall the Theorem 2 from Section 5.

Theorem 2. *Given a* swT *T over Σ, Δ_{i}, commutative $\mathbb{S}$, and $\bar{\Phi}$, and a* swVPA *A over Δ, $\mathbb{S}$, $\bar{\Phi}$, one can construct in PTIME a* swVPT *$T \otimes A$, over Σ, Δ, $\mathbb{S}$, $\bar{\Phi}$, such that $\forall s \in \Sigma^*$, $t \in \Delta^*$, $(T \otimes A)(s,t) = T(s, t|_{\Delta_{\mathsf{i}}}) \otimes A(t)$.*

Proof. Let $T = \langle Q_T, \mathsf{in}_T, \bar{\mathsf{w}}_T, \mathsf{out}_T\rangle$, where $\bar{\mathsf{w}}_T$ contains w_{10}, w_{01}, and w_{11}, and let $A = \langle Q_A, \mathsf{in}_A, \bar{\mathsf{w}}_A, \mathsf{out}_A\rangle$ where $\bar{\mathsf{w}}_A$ contains w_{i}, w_{c}, w_{r}, $\mathsf{w}^{\mathsf{e}}_{\mathsf{r}}$.

The set of states of $T \otimes A$ will be $Q' = Q_T \times Q_A$, and its set of stack symbols $P' = P$. The entering, leaving and transition functions of $T \otimes A$ will simulate the synchronized computations of T and A on respectively the pair $\langle s, t|_{\Delta_{\mathsf{i}}}\rangle$ and t, while reading a pair $\langle s,t\rangle \in \Sigma^* \times \Delta^*$. The state entering and leaving functions of $T \otimes A$ are defined, for all $\langle q_T, q_A\rangle \in Q'$, by:

$$\begin{aligned}
\mathsf{in}'\big(\langle q_T, q_A\rangle\big) &= \mathsf{in}_T(q_T) \otimes \mathsf{in}_A(q_A)\\
\mathsf{out}'\big(\langle q_T, q_A\rangle\big) &= \mathsf{out}_T(q_T) \otimes \mathsf{out}_A(q_A)
\end{aligned}$$

The transition functions of $T \otimes A$ are defined by:

$$\begin{aligned}
\mathsf{w}'_{10}\big(\langle q_T, q_A\rangle, \langle q'_T, q_A\rangle\big) &= \mathsf{w}_{10}(q_T, q'_T) & &\\
\mathsf{w}'_{10}\big(\langle q_T, q_A\rangle, \langle q'_T, q'_A\rangle\big) &= \mathbb{0} & &\text{when } q_A \neq q'_A\\
\mathsf{w}'_{01}(\langle q_T, q_A\rangle, \langle q_T, q'_A\rangle) &= \mathsf{w}_{\mathsf{i}}(q_A, q'_A) & &\\
\mathsf{w}'_{01}(\langle q_T, q_A\rangle, \langle q'_T, q'_A\rangle) &= \mathbb{0} & &\text{when } q_T \neq q'_T\\
\mathsf{w}'_{11}(\langle q_T, q_A\rangle, \langle q'_T, q'_A\rangle) &= \mathsf{w}_{11}(q_T, q'_T) \otimes_2 \mathsf{w}_{\mathsf{i}}(q_A, q'_A) & &\\
\mathsf{w}_{\mathsf{c}}'(\langle q_T, q_A\rangle, \langle q_T, q'_A\rangle, p) &= \mathsf{w}_{\mathsf{c}}(q_A, q'_A, p) & &\\
\mathsf{w}_{\mathsf{c}}'(\langle q_T, q_A\rangle, \langle q'_T, q'_A\rangle, p) &= \mathbb{0} & &\text{when } q_T \neq q'_T\\
\mathsf{w}_{\mathsf{r}}'(\langle q_T, q_A\rangle, p, \langle q_T, q'_A\rangle) &= \mathsf{w}_{\mathsf{r}}(q_A, p, q'_A) & &\\
\mathsf{w}_{\mathsf{r}}'(\langle q_T, q_A\rangle, p, \langle q'_T, q'_A\rangle) &= \mathbb{0} & &\text{when } q_T \neq q'_T\\
\mathsf{w}^{\mathsf{e}\prime}_{\mathsf{r}}(\langle q_T, q_A\rangle, \langle q_T, q'_A\rangle) &= \mathsf{w}^{\mathsf{e}}_{\mathsf{r}}(q_A, q'_A) & &\\
\mathsf{w}^{\mathsf{e}\prime}_{\mathsf{r}}(\langle q_T, q_A\rangle, \langle q'_T, q'_A\rangle) &= \mathbb{0} & &\text{when } q_T \neq q'_T
\end{aligned}$$

It means that, for all $e \in \Sigma$, $a \in \Delta_{\mathsf{i}}$, $c \in \Delta_{\mathsf{c}}$, $r \in \Delta_{\mathsf{r}}$:

$$\begin{aligned}
\mathsf{w}'_{10}\big(\langle q_T, q_A\rangle, e, \varepsilon, \langle q'_T, q_A\rangle\big) &= \phi(e) & &\text{where } \phi = \mathsf{w}_{10}(q_T, q'_T)\\
\mathsf{w}'_{10}\big(\langle q_T, q_A\rangle, e, \varepsilon, \langle q'_T, q'_A\rangle\big) &= \mathbb{0} & &\text{when } q_A \neq q'_A\\
\mathsf{w}'_{01}(\langle q_T, q_A\rangle, \varepsilon, a, \langle q_T, q'_A\rangle) &= \phi(a) & &\text{where } \phi = \mathsf{w}_{\mathsf{i}}(q_A, q'_A),\\
\mathsf{w}'_{01}(\langle q_T, q_A\rangle, \varepsilon, a, \langle q'_T, q'_A\rangle) &= \mathbb{0} & &\text{when } q_T \neq q'_T\\
\mathsf{w}'_{11}(\langle q_T, q_A\rangle, e, a, \langle q'_T, q'_A\rangle) &= \eta(e,a) \otimes \phi(a) & &\text{where } \eta = \mathsf{w}_{11}(q_T, q'_T),\\
& & &\phi = \mathsf{w}_{\mathsf{i}}(q_A, q'_A)
\end{aligned}$$

$$
\begin{aligned}
\mathsf{w_c}'(\langle q_T, q_A\rangle, \varepsilon, c, \langle q_T, q'_A\rangle, p) &= \phi(c) &&\text{where } \phi = \mathsf{w_c}(q_A, q'_A, p),\\
\mathsf{w_c}'(\langle q_T, q_A\rangle, \varepsilon, c, \langle q'_T, q'_A\rangle, p) &= \mathbb{0} &&\text{when } q_T \neq q'_T\\
\mathsf{w_r}'(\langle q_T, q_A\rangle, c, p, \varepsilon, r, \langle q_T, q'_A\rangle) &= \eta(c, r) &&\text{where } \eta = \mathsf{w_r}(q_A, p, q'_A)\\
\mathsf{w_r}'(\langle q_T, q_A\rangle, c, p, \varepsilon, r, \langle q'_T, q'_A\rangle) &= \mathbb{0} &&\text{when } q_T \neq q'_T\\
\mathsf{w_r^e}'(\langle q_T, q_A\rangle, \varepsilon, r, \langle q_T, q'_A\rangle) &= \phi(r) &&\text{where } \phi = \mathsf{w_r^e}(q_A, q'_A)\\
\mathsf{w_r^e}'(\langle q_T, q_A\rangle, \varepsilon, r, \langle q'_T, q'_A\rangle) &= \mathbb{0} &&\text{when } q_T \neq q'_T
\end{aligned}
$$

The transition w'_{10} performs their counterpart in T while reading one input symbol e, ignoring A: the state q_A of A is left unchanged in this transition, which does not read the output symbol. On the other hand, the transitions w'_{01}, $\mathsf{w_c}'$, $\mathsf{w_r}'$, and $\mathsf{w_r^e}'$ perform their counterparts in A while reading one output symbol a or c or r, and ignores T: the state q_T of T is unchanged in these transitions, and no input symbol is read. The transition w'_{11} simulates the computation of both T and A simultaneously, while reading one input symbol e and one output symbol a.

The proof of the correctness of the construction, i.e. that $\forall s \in \Sigma^*$, $t \in \Delta^*$, $(T \otimes A)(s, t) = T(s, t|_{\Delta_\mathsf{i}}) \otimes A(t)$, is a straighthforward double induction on the length of s and t. □

D Proof of Theorem 3

The swVPT of Definition 1 does not contain ε-transitions. However, this notion shall be convenient in the proof Theorem 3. It is defined formally as follows.

Definition 4 ($\mathsf{swVPT}_\varepsilon$). *A Symbolic Weighted Visibly Pushdown Transducer with ε-transitions over Σ, Δ, complete $\mathbb{S}$, and $\bar{\Phi}$ is a tuple $T = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$, where Q, P, in and out are like in Definition 1 and $\bar{\mathsf{w}}$ contains an additional function $\mathsf{w}_{00} : Q \times Q \to \mathbb{S}$.*

The function weight of a $\mathsf{swVPT}_\varepsilon$ is computed by adding (with $\oplus$) the weight of possible finite sequences ε-transitions w_{00}. Formally, for a $\mathsf{swVPT}_\varepsilon$ T, let $\mathsf{weight}^{\not\varepsilon}_T$ be the function defined for T by the equations (1) (for the case of swVPT without ε-transitions). Then weight_T is the function $[Q \times \Gamma^*] \times \Sigma^* \times \Delta^* \times [Q \times \Gamma^*]$ into $\mathbb{S}$, defined by, for $q, q' \in Q$, $\gamma, \gamma' \in \Gamma^*$, and $u \in \Sigma^*$, $v \in \Delta^*$:

$$
\begin{aligned}
\mathsf{weight}_T\big(q[\gamma], u, v, q'[\gamma']\big) &= \bigoplus_{\substack{q_0 \cdots q_n \in Q^* \\ q_0 = q}} \bigotimes_{i=0}^{n-1} \mathsf{w}_{00}(q_i, q_{i+1}) \\
&\qquad \otimes \mathsf{weight}^{\not\varepsilon}_T\big(q_n[\gamma], u, v, q'[\gamma']\big) \qquad (7)\\
\mathsf{weight}^{\not\varepsilon}_T\big(q[\gamma], \varepsilon, \varepsilon, q'[\gamma']\big) &= \mathbb{1} \text{ if } q = q', \gamma = \gamma' \text{ and } \mathbb{0} \text{ otherwise}\\
\mathsf{weight}^{\not\varepsilon}_T\big(q[\gamma], e\,u, \varepsilon, q'[\gamma']\big) &= \bigoplus_{q'' \in Q} \mathsf{w}_{10}(q, e, \varepsilon, q'') \otimes \mathsf{weight}_T\big(q''[\gamma], u, \varepsilon, q'[\gamma']\big)
\end{aligned}
$$

$$\begin{aligned}
\mathsf{weight}^{\not\varepsilon}_T\big(q[\gamma], \varepsilon, a\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{01}(q,\varepsilon,a,q'') \otimes \mathsf{weight}_T\big(q''[\gamma], \varepsilon, v, q'[\gamma']\big)\\
\mathsf{weight}^{\not\varepsilon}_T\big(q[\gamma], e\,u, a\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{10}(q,e,\varepsilon,q'') \otimes \mathsf{weight}_T\big(q''[\gamma], u, a\,v, q'[\gamma']\big)\\
&\oplus \bigoplus_{q''\in Q} \mathsf{w}_{01}(q,\varepsilon,a,q'') \otimes \mathsf{weight}_T\big(q''[\gamma], e\,u, v, q'[\gamma']\big)\\
&\oplus \bigoplus_{q''\in Q} \mathsf{w}_{11}(q,e,a,q'') \otimes \mathsf{weight}_T\big(q''[\gamma], u, v, q'[\gamma']\big)\\
\mathsf{weight}^{\not\varepsilon}_T\big(q[\gamma], u, c\,v, q'[\gamma']\big) &= \bigoplus_{\substack{q''\in Q\\ p\in P}} \mathsf{w}_{\mathsf{c}}(q,\varepsilon,c,q'',p) \otimes \mathsf{weight}_T\big(q''\begin{bmatrix}\langle c,p\rangle\\ \gamma\end{bmatrix}, u, v, q'[\gamma']\big)\\
\mathsf{weight}^{\not\varepsilon}_T\big(q\begin{bmatrix}\langle c,p\rangle\\ \gamma\end{bmatrix}, u, r\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}_{\mathsf{r}}\big(q,c,p,\varepsilon,r,q''\big) \otimes \mathsf{weight}_T\big(q''[\gamma], u, v, q'[\gamma']\big)\\
\mathsf{weight}^{\not\varepsilon}_T\big(q[\bot], u, r\,v, q'[\gamma']\big) &= \bigoplus_{q''\in Q} \mathsf{w}^{\mathsf{e}}_{\mathsf{r}}(q,\varepsilon,r,q'') \otimes \mathsf{weight}_T\big(q''[\bot], u, v, q'[\gamma']\big)
\end{aligned}$$

The hypothesis that $\mathbb{S}$ is complete ensures that the possibly infinite sum in the first equation of (7) is well defined. The next equations of (7) are the same as in (1) where the weight_T in the left-hand-side is replaced by $\mathsf{weight}^{\not\varepsilon}_T$. Note that the weight_T in the right-hand-side is not replaced by $\mathsf{weight}^{\not\varepsilon}_T$, meaning the ε-transition of w_{00} can be performed at any compution step.

Lemma 5. *For all swVPT$_\varepsilon$ T_ε over Σ, Δ, commutative, idempotent, and complete $\mathbb{S}$, and $\bar{\Phi}$, there exists one swVPT T over Σ, Δ, $\mathbb{S}$, and $\bar{\Phi}$, of size polynomial in the size of T_ε and effectively constructible in PTIME in the size of T_ε, such that for all $\langle s,t\rangle \in \Sigma^* \times \Delta^*$, $T(s,t) = T_\varepsilon(s,t)$.*

Proof. Let $T_\varepsilon = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$. We build $T = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}', \mathsf{out}'\rangle$; the construction of $\bar{\mathsf{w}}'$ and out' follows the line of the ε-removal algorithm of [App27].
For all $q, q' \in Q$, let

$$\ell_{00}(q,q') = \bigoplus_{\substack{q_0\ldots q_n\in Q^*\\ q_0=q, q_n=q'}} \bigotimes_{i=0}^{n-1} \mathsf{w}_{00}(q_i, q_{i+1})$$

Since by hypothesis, $\mathbb{S}$ is commutative and idempotent, it holds that:

Fact 1. For all $q, q' \in Q$, there exists one sequence $q_0 \ldots q_n \in Q^*$ *without repetition*, such that $q_0 = q$, $q_n = q'$, and $\ell_{00}(q,q') = \bigotimes_{i=0}^{n-1} \mathsf{w}_{00}(q_i, q_{i+1})$.

Therefore, we can pre-compute every $\ell_{00}(q,q')$ in at most $|Q|$ iterations, with a Viterbi algorithm [13] for finding a shortest path in the graph defined by w_{00}.
Let, for all $q' \in Q$,

$$\mathsf{out}'(q) = \ell_{00}(q,q') \otimes \mathsf{out}'(q')$$

and, for all $q, q' \in Q$,

$$\mathsf{w}'_{10}(q,q') = \bigoplus_{q'' \in Q} \ell_{00}(q,q'') \otimes \mathsf{w}'_{10}(q'',q')$$

and similarly for w_{01}, w_{11}, $\mathsf{w_c}$, $\mathsf{w_r}$, $\mathsf{w_r^e}$.

Fact 1, implies that: $\mathsf{out}'(q) = \bigoplus_{\substack{q_0 \ldots q_n \in Q^* \\ q_0 = q}} \bigotimes_{i=0}^{n-1} \mathsf{w}_{00}(q_i, q_{i+1}) \otimes \mathsf{out}(q_n)$

and $\mathsf{w}'_{10}(q,q') = \bigoplus_{\substack{q_0 \ldots q_n \in Q^* \\ q_0 = q}} \bigotimes_{i=0}^{n-1} \mathsf{w}_{00}(q_i, q_{i+1}) \otimes \mathsf{w}_{10}(q_n, q')$.

By (7), it follows that $T(s,t) = T_\varepsilon(s,t)$ for all $\langle s,t\rangle \in \Sigma^* \times \Delta^*$. $\square$

Theorem 3. *Given a swVPT T over Σ, Δ, commutative, complete and idempotent $\mathbb{S}$, and $\bar{\Phi}$, and given $s \in \Sigma^*$, there exists an effectively constructible swVPA $T(s)$ over Δ, $\mathbb{S}$, and $\bar{\Phi}$, such that for every $t \in \Delta^*$, $T(s)(t) = T(s,t)$.*

Proof. Let $T = \langle Q, P, \mathsf{in}, \bar{\mathsf{w}}, \mathsf{out}\rangle$, where $\bar{\mathsf{w}}$ contains w_{10}, w_{01}, and w_{11}, from $Q \times Q$ into respectively Φ_e, Φ_i, and Φ_ei, and $\mathsf{w_c} : Q \times Q \times P \to \Phi_\mathsf{c}$, $\mathsf{w_r} : Q \times P \times Q \to \Phi_\mathsf{cr}$, $\mathsf{w_r^e} : Q \times Q \to \Phi_\mathsf{r}$ and let $s = e_1 \ldots e_k$.
We construct a swVPA with ε-transitions $T_\varepsilon(s) = \langle Q', P', \mathsf{in}', \bar{\mathsf{w}}', \mathsf{out}'\rangle$, with a state set $Q' = [0..k] \times Q$, a set of stack symbols $P' = P$. The functions in', out' and $\bar{\mathsf{w}}'$, will simulate the synchronized computation of T on $\langle s,t\rangle$, while reading an output word $t \in \Delta^*$.
The state entering function of $T_\varepsilon(s)$ is defined by, for all $q \in Q$:

$$\begin{aligned} \mathsf{in}'\big(\langle 0,q\rangle\big) &= \mathsf{in}(q) \\ \mathsf{in}'\big(\langle i,q\rangle\big) &= \mathbb{0} \qquad \text{for } 0 < i \le k \end{aligned}$$

and the state leaving function is defined by, for all $q \in Q$:

$$\begin{aligned} \mathsf{out}'\big(\langle k,q\rangle\big) &= \mathsf{out}(q) \\ \mathsf{out}'\big(\langle i,q\rangle\big) &= \mathbb{0} \qquad \text{for } 0 \le i < k. \end{aligned}$$

Regarding transition functions of $\bar{\mathsf{w}}'$, for all $q, q' \in Q$,

$$\begin{aligned} \mathsf{w_i}'\big(\langle i,q\rangle, \langle i,q'\rangle\big) &= \mathsf{w}_{01}(q,q') && \text{for } 0 \le i \le k \\ \mathsf{w_i}'\big(\langle i,q\rangle, \langle i+1,q'\rangle\big) &: y \mapsto \mathsf{w}_{11}(q, e_i, y, q') && \text{for } 0 \le i < k \\ \mathsf{w_i}'\big(\langle i,q\rangle, \langle i',q'\rangle\big) &= \mathbb{0} && \text{for } 0 \le i, i' \le k, i' \ne i, i' \ne i+1. \end{aligned}$$

The ε-transitions of $T_\varepsilon(s)$ are, for all $q, q' \in Q$,

$$\begin{aligned} \mathsf{w}_{00}\big(\langle i,q\rangle, \langle i+1,q'\rangle\big) &= \mathsf{w}_{10}(q, e_i, \varepsilon, q') && \text{for } 0 \le i < k \\ \mathsf{w}_{00}\big(\langle i,q\rangle, \langle i',q'\rangle\big) &= \mathbb{0} && \text{for } 0 \le i, i' \le k, i' \ne i+1. \end{aligned}$$

And the other transitions of $T_\varepsilon(s)$ are, for all $q, q' \in Q$, $p \in P$,

$$\begin{array}{lll}
\mathsf{w_c}'\big(\langle i,q\rangle, \langle i,q'\rangle, p\big) = \mathsf{w_c}(q,q') & \text{for } 0 \le i \le k \\
\mathsf{w_c}'\big(\langle i,q\rangle, \langle i',q'\rangle, p\big) = \mathbb{0} & \text{for } 0 \le i, i' \le k, i' \neq i \\
\mathsf{w_r}'\big(\langle i,q\rangle, p, \langle i,q'\rangle\big) = \mathsf{w_r}(q,p,q') & \text{for } 0 \le i \le k \\
\mathsf{w_r}'\big(\langle i,q\rangle, p, \langle i',q'\rangle, \big) = \mathbb{0} & \text{for } 0 \le i, i' \le k, i' \neq i \\
\mathsf{w_r^e}'\big(\langle i,q\rangle, \langle i,q'\rangle\big) = \mathsf{w_r^e}(q,q') & \text{for } 0 \le i \le k \\
\mathsf{w_r^e}'\big(\langle i,q\rangle, \langle i',q'\rangle, \big) = \mathbb{0} & \text{for } 0 \le i, i' \le k, i' \neq i
\end{array}$$

We can show that for all $t \in \Delta^*$, $T_\varepsilon(s)(t) = T(s,t)$. Hence Theorem 3 follows, by Lemma 5. □

E End of proof of Theorem 4

Lemma 4 below shows that the computation of $b_\perp$, and by extension the computation of the minimum of A over Δ^*, reduces to the search of a best weighted derivation in the hypergraph $\mathcal{G}_A$, defined in Section 5.

Lemma 4. *For all $q, q' \in Q$, $b_\perp(q,q') = \bigoplus_{\theta \in \mathcal{D}(\mathcal{G}_A, \langle q,\perp,q'\rangle)} \mathsf{weight}(\theta)$.*

The direction $\le_\oplus$ of Lemma 4 follows from Lemma 6 below. In the following, we use the notation from Appendix C where we consider the weights of computations of swVPA as particular cases of swVPT, i.e. the argument in (1) that corresponds to an input symbol of Σ (for a swVPT) is ignored. Note the use of the special symbol $\top$ in configurations like $q[\top]$ in the expressions of weight_A. With such a symbol for γ in (1), the computation of weight_A is ensured to start with a non-empty stack, and never reads or pops the top of this stack.

Lemma 6 (Correctness). *For all derivation $\theta \in \mathcal{D}(\mathcal{G}_A, \langle q,\gamma,q'\rangle)$ such that $\mathsf{weight}(\theta) \neq \mathbb{0}$, where $\gamma \in \{\perp, \top\}$ and $q, q' \in Q$, there exists a word $t \in \Delta^*$ such that $\mathsf{weight}_A(q[\gamma], t, q'[\gamma]) = \mathsf{weight}(\theta)$.*

Proof. By induction on the size of the derivation θ rooted by $\langle q,\gamma,q'\rangle$.
The base case is when θ is composed of a single vertex. In order to ensure that $\mathsf{weight}(\theta) \neq \mathbb{0}$, this vertex shall have the form $\langle q,\perp,q\rangle$ or $\langle q,\top,q\rangle$. In both cases, $\mathsf{weight}(\theta) = \mathbb{1}$, and by (1), $\mathsf{weight}_A(q[\perp], \varepsilon, q'[\perp]) = \mathsf{weight}_A(q[\top], \varepsilon, q'[\top]) = \mathbb{1}$. Hence the property holds with $t = \varepsilon$.

If $\theta = v(\theta_1)$, where $\theta_1 \in \mathcal{D}(\mathcal{G}_A, v_1)$, let us assume that Lemma 6 holds for θ_1, and a word $t_1 \in \Delta^*$. We do a case analysis on the hyperedge $\langle v_1, v\rangle$.

Firstly, let us consider the case where $v_1 = \langle q_0,\perp,q_1\rangle$ and $v = \langle q_0,\perp,q_2\rangle$ for some $q_0, q_1, q_2 \in Q$. By the hypothesis that $\mathbb{S}$ is total, we are in one of the following two cases:

– $\eta_A(v_1, v) = \bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q_1, q_2)$. By effectiveness of $\bar{\Phi}$, there exists $a \in \Delta_\mathsf{i}$ such that $\bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q_1, q_2) = \mathsf{w_i}(q_1, a, q_2)$. It follows that:

$$\begin{aligned}
\mathsf{weight}(\theta) &= \mathsf{weight}(\theta_1) \otimes \eta_A(v_1, v) \\
&= \mathsf{weight}(\theta_1) \otimes \mathsf{w_i}(q_1, a, q_2) \\
&= \mathsf{weight}_A(q_0[\bot], t_1, q_1[\bot]) \otimes \mathsf{w_i}(q_1, a, q_2) \text{ by induction hypothesis} \\
&= \mathsf{weight}_A(q_0[\bot], t_1\, a, q_2[\bot]) \\
&\quad \text{by (1) and associativity, commutativity of } \otimes,
\end{aligned}$$

and the lemma holds with $t = t_1\, a$.

– $\eta_A(v_1, v) = \bigoplus_{\Delta_\mathsf{r}} \mathsf{w_r^e}(q_1, q_2)$, and we can proceed similarly as above in order to find $t = t_1\, r$ as expected, for some $r \in \Delta_\mathsf{r}$ (case of an unmatched return symbol).

Secondly, we consider the case where $v_1 = \langle q_1, \top, q_2\rangle$ and $v = \langle q_0, \bot, q_3\rangle$ for $q_0, q_1, q_2, q_3 \in Q$. In this case,

$$\eta_A(v_1, v) = \bigoplus_{p \in P} \bigoplus_{\Delta_\mathsf{c}} \big[\mathsf{w_c}(q_0, q_1, p) \otimes \bigoplus^2_{\Delta_\mathsf{r}} \mathsf{w_r}(q_2, p, q_3)\big].$$

By hypothesis, this value is not $\mathbb{0}$, hence there exists a stack symbol $p \in P$, a call symbol $c \in \Delta_\mathsf{c}$, and a return symbol $r \in \Delta_\mathsf{r}$ such that $\eta_A(v_1, v) = \mathsf{w_c}(q_0, c, q_1, p) \otimes \mathsf{w_r}(q_2, c, p, r, q_3)$. Therefore,

$$\begin{aligned}
\mathsf{weight}(\theta) &= \mathsf{weight}(\theta_1) \otimes \eta_A(v_1, v) \\
&= \mathsf{weight}(\theta_1) \otimes \mathsf{w_c}(q_0, c, q_1, p) \otimes \mathsf{w_r}(q_2, c, p, r, q_3) \\
&= \mathsf{weight}_A(q_1[\top], t_1, q_2[\top]) \otimes \mathsf{w_c}(q_0, c, q_1, p) \otimes \mathsf{w_r}(q_2, c, p, r, q_3) \\
&\quad \text{by induction hypothesis} \\
&= \mathsf{weight}_A(q_0[\bot], c\, t_1\, r, q_3[\bot]) \\
&\quad \text{by (1) and associativity, commutativity of } \otimes,
\end{aligned}$$

and we can conclude with $t = c\, t_1\, r$.

Finally, let us consider the case where $v_1 = \langle q_1, \top, q_2\rangle$ and $v = \langle q_0, \top, q_3\rangle$ for $q_0, q_1, q_2, q_3 \in Q$. Since $\mathbb{S}$ is total, there are two cases for the value of $\eta_A(v_1, v)$.

– $\eta_A(v_1, v) = \bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q_2, q_3)$ and $q_1 = q_0$. By effectiveness of $\bar{\Phi}$, there exists $a \in \Delta_\mathsf{i}$ such that $\bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q_2, q_3) = \mathsf{w}_{01}(q_2, \varepsilon, a, q_3) = \mathsf{w_i}(q_2, a, q_3)$, and

$$\begin{aligned}
\mathsf{weight}(\theta) &= \mathsf{weight}(\theta_1) \otimes \eta_A(v_1, v) \\
&= \mathsf{weight}(\theta_1) \otimes \mathsf{w_i}(q_2, a, q_3) \\
&= \mathsf{weight}_A(q_0[\top], t_1, q_2[\top]) \otimes \mathsf{w_i}(q_2, a, q_3) \text{ by induction hypothesis} \\
&= \mathsf{weight}_A(q_0[\top], t_1\, a, q_3[\top]),
\end{aligned}$$

and the lemma holds with $t = t_1\, a$.

– $\eta_A(v_1, v) = \bigoplus_{p \in P} \bigoplus_{\Delta_\mathsf{c}} \big[\mathsf{w_c}(q_0, q_1, p) \otimes \bigoplus^2_{\Delta_\mathsf{r}} \mathsf{w_r}(q_2, p, q_3)\big]$. Since this value is not $\mathbb{0}$ by hypothesis, there exists a stack symbol $p \in P$, a call symbol $c \in$

Δ_{c}, and a return symbol $r \in \Delta_{\mathsf{r}}$ such that $\eta_A(v_1, v) = \mathsf{w}_{\mathsf{c}}(q_0, c, q_1, p) \otimes \mathsf{w}_{\mathsf{r}}(q_2, c, p, r, q_3)$, and,

$$\begin{aligned}
\mathsf{weight}(\theta) &= \mathsf{weight}(\theta_1) \otimes \eta_A(v_1, v)\\
&= \mathsf{weight}(\theta_1) \otimes \mathsf{w}_{\mathsf{c}}(q_0, c, q_1, p) \otimes \mathsf{w}_{\mathsf{r}}(q_2, c, p, r, q_3)\\
&= \mathsf{weight}_A(q_1[\top], t_1, q_2[\top]) \otimes \mathsf{w}_{\mathsf{c}}(q_0, c, q_1, p) \otimes \mathsf{w}_{\mathsf{r}}(q_2, c, p, r, q_3)\\
&\quad \text{by induction hypothesis}\\
&= \mathsf{weight}_A(q_0[\top], c\, t_1\, r, q_3[\top])\\
&\quad \text{by (1) and associativity, commutativity of } \otimes,
\end{aligned}$$

and the lemma holds with $t = c\, t_1\, r$.

If $\theta = v(\theta_1, \theta_2)$, where $\theta_1 \in \mathcal{D}(\mathcal{G}_A, v_1)$ and $\theta_2 \in \mathcal{D}(\mathcal{G}_A, v_2)$, for vertices $v_1, v_2 \in V_A$, let us assume that Lemma 6 holds, on the one side for θ_1 and a word $t_1 \in \Delta^*$, on the other side for θ_2 and a word $t_2 \in \Delta^*$. Since by hypothesis $\mathsf{weight}(\theta) = \mathsf{weight}(\theta_1) \otimes \mathsf{weight}(\theta_2) \otimes \eta_A(v_1, v_2, v) \neq \mathbb{0}$, it holds that $\eta_A(v_1, v_2, v) = \mathbb{1}$. Hence, by construction, $v_1 = \langle q_0, \gamma, q_1 \rangle$, $v_2 = \langle q_1, \gamma, q_2 \rangle$, and $v = \langle q_0, \gamma, q_2 \rangle$, for some for $q_0, q_1, q_2 \in Q$ and $\gamma \in \{\perp, \top\}$. Then,

$$\begin{aligned}
\mathsf{weight}(\theta) &= \mathsf{weight}(\theta_1) \otimes \mathsf{weight}(\theta_2) \otimes \eta_A(v_1, v_2, v)\\
&= \mathsf{weight}(\theta_1) \otimes \mathsf{weight}(\theta_2)\\
&= \mathsf{weight}_A(q_0[\gamma], t_1, q_1[\gamma]) \otimes \mathsf{weight}_A(q_1[\gamma], t_2, q_2[\gamma])\\
&\quad \text{by induction hypothesis}\\
&= \mathsf{weight}_A(q_0[\gamma], t_1\, t_2, q_2[\gamma]) \text{ by (1),}
\end{aligned}$$

and we can conclude the lemma with $t = t_1\, t_2$. □

The direction $\geq_\oplus$ of Lemma 4 follows from the Lemma 7 below. In this lemma, we call a word $t \in \Delta^*$ *well-parenthesised* if it is either:

$t = \varepsilon$, the empty word, or
$t = t_1\, a$, for $a \in \Delta_{\mathsf{i}}$ and some well-parenthesised word t_1, or
$t = t_1\, r$, for $r \in \Delta_{\mathsf{r}}$, and some well-parenthesised word t_1, or
$t = c\, t_1\, r$, for $r \in \Delta_{\mathsf{r}}$, $c \in \Delta_{\mathsf{c}}$, and some well-parenthesised word t_1, or
$t = t_1 t_2$, for some well-parenthesised words t_1, t_2.

Lemma 7 (Completeness). *For all well-parenthesised $t \in \Delta^*$, for all $q, q' \in Q$, and $\gamma \in \{\perp, \top\}$, there exists a derivation $\theta \in \mathcal{D}\big(\mathcal{G}_A, \langle q, \gamma, q' \rangle\big)$ such that $\mathsf{weight}(\theta) \leq_\oplus \mathsf{weight}_A(q[\gamma], t, q'[\gamma])$.*

Proof. By induction on the length of t. If the length of t is zero, then by (1), $\mathsf{weight}_A(q[\gamma], t, q'[\gamma]) = \mathbb{1}$ if $q = q'$ and $\mathsf{weight}_A(q[\gamma], t, q'[\gamma]) = \mathbb{0}$ otherwise. In both cases, we can choose the singleton derivation $\theta = \langle q, \gamma, q' \rangle$.
Let us now assume that the length of t is strictly greater than 0. Since t is well-parenthesised by hypothesis, we are in one of the following four cases.

If $t = t_1\, a$, for $a \in \Delta_{\mathsf{i}}$, and some well-parenthesised word t_1, by (1), it holds that

$$\mathsf{weight}_A(q[\gamma], t, q'[\gamma]) = \mathsf{weight}_A(q[\gamma], t_1, q''[\gamma]) \otimes \mathsf{w}_{\mathsf{i}}(q'', a, q')$$

for some $q'' \in Q$. By induction hypothesis, there exists a derivation $\theta_1 \in \mathcal{D}\big(\mathcal{G}_A, \langle q, \gamma, q''\rangle\big)$ such that: $\mathsf{weight}(\theta_1) \leq_\oplus \mathsf{weight}_A(q[\gamma], t_1, q''[\gamma])$. Let $\theta = \langle q, \gamma, q'\rangle(\theta_1)$. If $\gamma = \bot$, then we have:

$$\mathsf{weight}(\theta) = \mathsf{weight}(\theta_1) \otimes \Big[\bigoplus\nolimits_{\Delta_\mathsf{i}} \mathsf{w_i}(q'', q') \oplus \bigoplus\nolimits_{\Delta_\mathsf{r}} \mathsf{w_r^e}(q'', q')\Big].$$

By monotony of $\mathbb{S}$ (Lemma 1), $\bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q'', q') \oplus \bigoplus_{\Delta_\mathsf{r}} \mathsf{w_r^e}(q'', q') \leq_\oplus \mathsf{w_i}(q'', a, q')$. If $\gamma = \top$, then:

$$\mathsf{weight}(\theta) = \mathsf{weight}(\theta_1) \otimes \left[\begin{array}{l} \bigoplus_{\Delta_\mathsf{i}} \mathsf{w_i}(q'', q') \\ \oplus \displaystyle\bigoplus_{p \in P} \bigoplus\nolimits_{\Delta_\mathsf{c}} \big[\mathsf{w_c}(q, q, p) \otimes_2 \bigoplus\nolimits^2_{\Delta_\mathsf{r}} \mathsf{w_r}(q'', p, q')\big] \end{array}\right]$$

and again, by Lemma 1, $F_2 \leq_\oplus \mathsf{w_i}(q'', a, q')$ where F_2 is the second factor of the above expression. Altogether, it holds that:

$$\begin{aligned} \mathsf{weight}(\theta) &\leq_\oplus \mathsf{weight}(\theta_1) \otimes \mathsf{w_i}(q'', a, q') \\ &\leq_\oplus \mathsf{weight}_A(q[\gamma], t_1, q''[\gamma]) \otimes \mathsf{w_i}(q'', a, q') = \mathsf{weight}_A(q[\gamma], t, q'[\gamma]). \end{aligned}$$

If $t = t_1\, r$, for $r \in \Delta_\mathsf{r}$, and some well-parenthesised word t_1: the proof is similar to the above case.

If $t = c\, t_1\, r$ for $c \in \Delta_\mathsf{c}$, $r \in \Delta_\mathsf{r}$, and some well-parenthesised word t_1, we have, by (1), for some $p \in P$ and some $q_1, q_2 \in Q$:

$$\mathsf{weight}_A(q[\gamma], t, q'[\gamma]) = \mathsf{w_c}(q, c, q_1, p) \otimes \mathsf{weight}_A(q_1[\top], t_1, q_2[\top]) \otimes \mathsf{w_r}(q_2, c, p, r, q').$$

Note that in the intermediate computation from q_1 to q_2, the stack must not be empty, because it contains at least the pair $\langle c, p\rangle$ on top.
By induction hypothesis, there exists a derivation $\theta_1 \in \mathcal{D}\big(\mathcal{G}_A, \langle q_1, \top, q_2\rangle\big)$ such that $\mathsf{weight}(\theta_1) \leq_\oplus \mathsf{weight}_A(q_1[\top], t_1, q_2[\top])$. Let $\theta = \langle q, \gamma, q'\rangle(\theta_1)$. It holds that:

$$\mathsf{weight}(\theta) = \mathsf{weight}(\theta_1) \otimes \eta_A(\langle q_1, \top, q_2\rangle, \langle q, \gamma, q'\rangle).$$

The hyperedge's weight $H = \eta_A(\langle q_1, \top, q_2\rangle, \langle q, \gamma, q'\rangle)$ can take one of the following two values:

$$\text{if } \gamma = \bot,\ H = \bigoplus_{p' \in P} \bigoplus\nolimits_{\Delta_\mathsf{c}} \big[\mathsf{w_c}(q, q_1, p') \otimes \bigoplus\nolimits^2_{\Delta_\mathsf{r}} \mathsf{w_r}(q_2, p', q')\big],$$

$$\text{if } \gamma = \top,\ H = \Big[\bigoplus_{q_1 = q} \bigoplus\nolimits_{\Delta_\mathsf{i}} \mathsf{w_i}(q_2, q')\Big] \oplus \Big[\bigoplus_{p' \in P} \bigoplus\nolimits_{\Delta_\mathsf{c}} \big[\mathsf{w_c}(q, q_1, p') \otimes_2 \bigoplus\nolimits^2_{\Delta_\mathsf{r}} \mathsf{w_r}(q_2, p', q')\big]\Big].$$

By Lemma 1, in both cases, it holds that:

$$\eta_A(\langle q_1, \top, q_2\rangle, \langle q, \gamma, q'\rangle) \leq_\oplus \mathsf{w_c}(q, c, q_1, p) \otimes \mathsf{w_r}(q_2, c, p, r, q').$$

Therefore,

$$\begin{aligned} \mathsf{weight}(\theta) &\leq_\oplus \mathsf{w_c}(q, c, q_1, p) \otimes \mathsf{weight}(\theta_1) \otimes \mathsf{w_r}(q_2, c, p, r, q') \\ &\leq_\oplus \mathsf{w_c}(q, c, q_1, p) \otimes \mathsf{weight}_A(q_1[\top], t_1, q_2[\top]) \otimes \mathsf{w_r}(q_2, c, p, r, q') \\ &\leq_\oplus \mathsf{weight}_A(q[\gamma], t, q'[\gamma]). \end{aligned}$$

Finally, if $t = t_1 t_2$ for t_1, t_2 two well-parenthesised words, we have,

$$\mathsf{weight}_A(q[\gamma], t, q'[\gamma]) = \mathsf{weight}_A(q[\gamma], t_1, q''[\gamma]) \otimes \mathsf{weight}_A(q''[\gamma], t_2, q'[\gamma])$$

for some $q'' \in Q$. The state γ of the stack is the same at the beginning and the end of the computation on t_1 (resp. t_2) because this word is well-parenthesised. By induction hypothesis, there exist derivations $\theta_1 \in \mathcal{D}\big(\mathcal{G}_A, \langle q, \gamma, q'' \rangle\big)$ and $\theta_2 \in \mathcal{D}\big(\mathcal{G}_A, \langle q'', \gamma, q' \rangle\big)$ such that $\mathsf{weight}(\theta_1) \leq_\oplus \mathsf{weight}_A(q[\gamma], t_1, q''[\gamma])$, and $\mathsf{weight}(\theta_2) \leq_\oplus \mathsf{weight}_A(q''[\gamma], t_2, q'[\gamma])$.
Let $\theta = \langle q, \gamma, q' \rangle(\theta_1, \theta_2)$. It holds that:

$$\begin{aligned}
\mathsf{weight}(\theta) &= \mathsf{weight}(\theta_1) \otimes \mathsf{weight}(\theta_2) \otimes \eta_A(\langle q, \gamma, q'' \rangle, \langle q'', \gamma, q' \rangle, \langle q, \gamma, q' \rangle) \\
&= \mathsf{weight}(\theta_1) \otimes \mathsf{weight}(\theta_2) \\
&\leq_\oplus \mathsf{weight}_A(q[\gamma], t_1, q''[\gamma]) \otimes \mathsf{weight}_A(q''[\gamma], t_2, q'[\gamma]) \\
&\leq_\oplus \mathsf{weight}_A(q[\gamma], t, q'[\gamma]).
\end{aligned}$$

□

We can now complete the proof of Lemma 4, and Theorem 4. Let $q, q' \in Q$.
If $\bigoplus_{\theta \in \mathcal{D}(\mathcal{G}_A, \langle q, \perp, q' \rangle)} \mathsf{weight}(\theta) = \mathbb{0}$, then for all derivation $\theta \in \mathcal{D}(\mathcal{G}_A, \langle q, \perp, q' \rangle)$, $\mathsf{weight}(\theta) = \mathbb{0}$, since this sum is finite and $\mathbb{S}$ is assumed total. Lemma 7 implies that for all $t \in \Delta^*$, $\mathsf{weight}_A(q[\perp], t, q'[\perp]) = \mathbb{0}$. Therefore, $b_\perp(q, q') = \mathbb{0}$ in this case.
Let us now assume that

$$\bigoplus_{\theta \in \mathcal{D}(\mathcal{G}_A, \langle q, \perp, q' \rangle)} \mathsf{weight}(\theta) = W \neq \mathbb{0}. \tag{8}$$

There exists $\theta_{q,q'} \in \mathcal{D}(\mathcal{G}_A, \langle q, \perp, q' \rangle)$, such that $W = \mathsf{weight}(\theta_{q,q'})$. By Lemma 6, there exists $t_{q,q'} \in \Delta^*$ such that $\mathsf{weight}_A(q[\perp], t_{q,q'}, q'[\perp]) = \mathsf{weight}(\theta_{q,q'}) = W$.
We can show by contradiction that for all $t \in \Delta^*$, $W \leq_\oplus \mathsf{weight}_A(q[\perp], t, q'[\perp])$. Indeed, assume on the opposite that $\mathsf{weight}_A(q[\perp], t, q'[\perp]) <_\oplus W$ for some $t \in \Delta^*$. Since this weight is computed by starting and ending with an empty stack, t is well-parenthesised, and by Lemma 7, there exists a derivation $\theta \in \mathcal{D}(\mathcal{G}_A, \langle q, \perp, q' \rangle)$, and $\mathsf{weight}(\theta) <_\oplus W$, contradicting (8).
Therefore, $b_\perp(q, q') = \mathsf{weight}_A(q[\perp], t_{q,q'}, q'[\perp]) = W$.
Moreover, the above word $t_{q,q'}$ is a witness reaching the minimum of the swVPA A computed by Theorem 4.

F Nested Words and Parse Trees

The hierarchical structure of nested words, defined with the *call* and *return* markup symbols, suggests a correspondence with trees. The lifting of this correspondence to languages, of tree automata and VPA, has been discussed in [2], and [6] for the weighted case. In this section, we describe a correspondence between the symbolic-weighted extensions of tree automata and VPA. It might be folklore knowledge but we state it explicitly for the sake of clarity.

Let Ω be a countable ranked alphabet, such that every symbol $a \in \Omega$ has a rank $\mathsf{rk}(a) \in [0..M]$ where M is a fixed natural number. We denote by Ω_k the subset of all symbols $a \in \Omega$ with $\mathsf{rk}(a) = k$, where $0 \leq k \leq M$, and $\Omega_{>0} = \Omega \setminus \Omega_0$. The free Ω-algebra of finite, ordered, Ω-labeled trees is denoted by $\mathcal{T}_\Omega$. It is the smallest set such that $\Omega_0 \subset \mathcal{T}_\Omega$, and, for all $1 \leq k \leq M$, all $a \in \Omega_k$, and all $t_1, \ldots, t_k \in \mathcal{T}_\Omega$, $a(t_1, \ldots, t_k) \in \mathcal{T}_\Omega$. Let us assume a commutative semiring $\mathbb{S}$ and a label theory $\bar{\Phi}$ over $\mathbb{S}$ containing one set Φ_{Ω_k} for each $k \in [0..M]$.

Definition 5. *A* symbolic-weighted tree automaton *(swTA) over Ω, $\mathbb{S}$, and $\bar{\Phi}$ is a triplet $A = \langle Q, \mathsf{in}, \bar{\mathsf{w}} \rangle$ where Q is a finite set of states, $\mathsf{in} : Q \to \mathbb{S}$ is the starting weight function, and $\bar{\mathsf{w}}$ is a tuplet of transition functions containing, for each $k \in [0..M]$, the function $\mathsf{w}_k : Q \times Q^k \to \Phi_{\Omega_k}$.*

We define a transition function $\mathsf{w} : Q \times \Omega \times \bigcup_{k=0}^{M} Q^k \to \mathbb{S}$ by ($q_1 \ldots q_k$ is ε if $k = 0$):

$$\mathsf{w}(q_0, b, q_1 \ldots q_k) = \phi(b) \quad \text{where } \phi = \mathsf{w}_k(q_0, q_1 \ldots q_k).$$

Every swTA defines a mapping from trees of $\mathcal{T}_\Omega$ into $\mathbb{S}$, based on the following intermediate function $\mathsf{weight}_A : Q \times \mathcal{T}_\Omega \to \mathbb{S}$

$$\mathsf{weight}_A(q_0, t) = \bigoplus_{q_1 \ldots q_k \in Q^k} \mathsf{w}(q_0, b, q_1 \ldots q_k) \otimes \bigotimes_{i=1}^{k} \mathsf{weight}_A(q_i, t_i) \tag{9}$$

where $q_0 \in Q$, and $t = b(t_1, \ldots, t_k) \in \mathcal{T}_\Omega$, with $0 \leq k \leq M$ (by convention, the product from 1 to k is equal to $\mathbb{1}$ when $k = 0$).

Finally, the weight associated by A to $t \in \mathcal{T}_\Omega$ is

$$A(t) = \bigoplus_{q \in Q} \mathsf{in}(q) \otimes \mathsf{weight}_A(q, t) \tag{10}$$

Intuitively, $\mathsf{w}(q_0, b, q_1 \ldots q_k)$ can be seen as the weight of a production rule $q_0 \to b(q_1, \ldots, q_k)$ of a regular tree grammar [App25], that replaces the non-terminal symbol q_0 by $b(q_1, \ldots, q_k)$. The above production rule can also be seen as a rule of a weighted CF grammar, of the form $[b]\, q_0 := q_1 \ldots q_k$ if $k > 0$, and $[b]\, q_0 := b$ if $k = 0$. In the first case, b is a label for the rule, and in the second case, it is also a terminal symbol. The weight of a labeled derivation tree t of the weighted CF grammar associated to A as above, is $\mathsf{weight}_A(q, t)$, when q is the start non-terminal.

We shall now establish a correspondence between such a derivation tree t and some word describing a linearization of t, in a way that $\mathsf{weight}_A(q, t)$ can be computed by a swVPA. Let $\hat{\Omega}$ be the countable (unranked) alphabet obtained from Ω by: $\hat{\Omega} = \Delta_{\mathsf{i}} \uplus \Delta_{\mathsf{c}} \uplus \Delta_{\mathsf{r}}$, with $\Delta_{\mathsf{i}} = \Omega_0$, $\Delta_{\mathsf{c}} = \{ \langle_a| \; a \in \Omega_{>0} \}$, $\Delta_{\mathsf{r}} = \{ {}_a\rangle \mid a \in \Omega_{>0} \}$. We associate to $\hat{\Omega}$ a label theory $\hat{\Phi}$ like in Section 3, and we define a linearization of trees of $\mathcal{T}_\Omega$ into words of $\hat{\Omega}^*$ as follows:

$\mathsf{lin}(a) = a$ for all $a \in \Omega_0$,
$\mathsf{lin}\big(b(t_1, \ldots, t_k)\big) = \langle_b \; \mathsf{lin}(t_1) \ldots \mathsf{lin}(t_k) \; {}_b\rangle$ when $b \in \Omega_k$ for $1 \leq k \leq M$.

Example 6. The trees in Figure 1 represent the two scores in Example 1, where we showed that their linearizations are respectively O and O'. $\diamond$

Proposition 2. *For all* swTA A *over* Ω, $\mathbb{S}$ *commutative, and* $\bar{\Phi}$, *there exists an effectively constructible* swVPA A' *over* $\hat{\Omega}$, $\mathbb{S}$ *and* $\hat{\Phi}$ *such that for all* $t \in \mathcal{T}_\Omega$, $A'\big(\mathsf{lin}(t)\big) = A(t)$.

Proof. We follow Definition 3 in Appendix C for swVPA, i.e. $\mathsf{w_i} = \mathsf{w}_{01}$ and we ignore the first symbol argument (input symbol) of Definition 1. Let $A = \langle Q, \mathsf{in}, \bar{\mathsf{w}}\rangle$ where $\bar{\mathsf{w}}$ is presented as above by a function. We build $A' = \langle Q', P', \mathsf{in}', \bar{\mathsf{w}}', \mathsf{out}'\rangle$, where $Q' = \bigcup_{k=0}^{M} Q^k$ is the set of sequences of state symbols of A, of length at most M, including the empty sequence denoted by ε, and where $P' = Q'$ and $\bar{\mathsf{w}}'$ is defined by ($\bar{u}, \bar{q} \in Q'$, $\bar{p} \in P'$):

$$\begin{array}{lll}
\mathsf{w_i}(q_0\,\bar{u}, a, \bar{u}) & = \mathsf{w}(q_0, a, \varepsilon) & \text{for all } a \in \Omega_0 \\
\mathsf{w_c}(q_0\,\bar{u}, \langle_c, \bar{q}, \bar{u}) & = \mathsf{w}(q_0, \langle_c, \bar{q}) & \text{for all } c \in \Omega_{>0} \\
\mathsf{w_r}(\varepsilon, \langle_c, \bar{p}, {}_c\rangle, \bar{p}) & = \mathbb{1} & \text{for all } c \in \Omega_{>0} \\
\mathsf{w_r^e}(\bar{u}, {}_c\rangle, \bar{q}) & = \mathbb{0} & \text{for all } c \in \Omega_{>0}
\end{array}$$

All cases not matched by one of the above equations have a weight $\mathbb{0}$, for instance $\mathsf{w_r}(\bar{u}, \langle_c, \bar{p}, {}_d\rangle, \bar{q}) = \mathbb{0}$ if $c \neq d$ or $\bar{u} \neq \varepsilon$ or $\bar{q} \neq \bar{p}$.
The entering and leaving weight functions $\mathsf{in}', \mathsf{out}' : Q' \to \mathbb{S}$ are defined by:

- $\mathsf{in}'(q) = \mathsf{in}(q)$ for all $q \in Q$,
- $\mathsf{in}'(\bar{q}) = \mathbb{0}$ for every other (non-singleton) sequence $\bar{q} \in Q'$,
- $\mathsf{in}'(\bar{q}) = \mathbb{1}$ for all $\bar{q} \in Q'$.

$\square$

References for appendices

25. Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Löding, C., Lugiez, D., Tison, S., Tommasi, M.: Tree Automata Techniques and Applications. `http://tata.gforge.inria.fr` (2007)
26. Huang, L.: Advanced dynamic programming in semiring and hypergraph frameworks. In: COLING (2008)
27. Lombardy, S., Sakarovitch, J.: The removal of weighted ε-transitions. In: International Conference on Implementation and Application of Automata. pp. 345–352. Springer (2012)