Context-Free Recognition with Weighted Automata

Corinna Cortes and **Mehryar Mohri**
{corinna,mohri}@research.att.com
AT&T Labs – Research
180 Park Avenue, Rm E147
Florham Park, NJ 07932, USA

Abstract

We introduce the definition of language recognition with weighted automata, a generalization of the classical definition of recognition with unweighted acceptors. We show that, with our definition of recognition, weighted automata can be used to recognize a class of languages that strictly includes regular languages. The class of languages accepted depends on the weight set which has the algebraic structure of a semiring. We give a generic linear time algorithm for recognition with weighted automata and describe examples with various weight sets illustrating the recognition of several classes of context-free languages. We prove, in particular, that the class of languages equivalent to the language of palindromes can be recognized by weighted automata over the $(+, \cdot)$-semiring, and that the class of languages equivalent to the Dyck language of first order $D_1'^*$ can be recognized by weighted automata over the real tropical semiring.

1 Introduction

Finite automata are used in many applications to build high-performance tools. But the recognition power of finite automata is limited to regular languages (Kleene, 1956). Many applications require more powerful devices to describe context-free or context-sensitive languages. We show that *weighted automata* can be used as such devices.

Our study of language recognition with weighted automata is motivated by several observations. First, weighted automata are currently used successfully in many applications such as text and speech processing (Mohri, 1997). Secondly, our definition of language recognition with weighted automata does not require writing new code: exactly the same algorithms as those used for manipulating and combining automata and transducers in applications such as speech processing can be used for recognition of context-free languages with weighted automata. Finally, these algorithms are based on the general theory of rational power series, which can be realized by weighted automata (Schützenberger, 1961; Eilenberg, 1974; Berstel and Reutenauer, 1988). Investigating the recognition power of weighted automata is equivalent to determining that of rational power series.

We introduce the definition of language recognition with weighted automata and show that it can be used to recognize a class of languages that strictly includes regular languages. The main idea behind the use of weighted automata for recognition of context-free languages is to exploit the additional information paths weights or multiplicities contain, just as stacks store additional information in the case of pushdown-automata.

The class of languages accepted depends on the weight set which has the algebraic structure of a semiring. We give a generic linear time algorithm for recognition with weighted automata and describe examples with various weight sets illustrating the recognition of several classes of context-free languages. We prove, in particular, that the class of languages equivalent to the language of palindromes can be recognized by weighted automata over the $(+, \cdot)$-semiring, and that the class of languages equivalent to the Dyck language of first order $D_1'^*$ can be recognized by weighted automata over the tropical semiring.

2 Weighted automata

Weighted automata are more general devices than unweighed automata in that their transitions are labeled with weights in addition to the usual alphabet symbols. For various operations to be well-defined, the weight set needs to have the algebraic structure of a semiring.

Definition 1 *A system* $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ *is a* right semiring *if:*

1. $(\mathbb{K}, \oplus, \overline{0})$ *is a commutative monoid with* $\overline{0}$ *as the identity element for* $\oplus$*,*
2. $(\mathbb{K}, \otimes, \overline{1})$ *is a monoid with* $\overline{1}$ *as the identity element for* $\otimes$*,*
3. $\otimes$ *right distributes over* $\oplus$*:* $\forall a, b, c \in \mathbb{K}, (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$*,*
4. $\overline{0}$ *is an annihilator for* $\otimes$*:* $\forall a \in \mathbb{K}, a \otimes \overline{0} = \overline{0} \otimes a = \overline{0}$*.*

Left semirings are defined in a similar way by replacing right distributivity with left distributivity. $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ is a *semiring* if both left and right distributivity hold. As an example, $(\mathbb{N}, +, \cdot, 0, 1)$ is a semiring defined on the set of nonnegative integers $\mathbb{N}$.

A *Weighted automaton* over the semiring $\mathbb{K}$ is a 7-tuple $A = (\Sigma, Q, I, F, E, \lambda, \rho)$ where Σ is the finite alphabet of the automaton, Q is a finite set of states, $I \subseteq Q$ the set of initial states, $F \subseteq Q$ the set of final states, $E \subseteq Q \times \Sigma \times \mathbb{K} \times Q$ a finite set of transitions, $\lambda : I \rightarrow \mathbb{K}$ the initial weight function mapping I to $\mathbb{K}$, and $\rho : F \rightarrow \mathbb{K}$ the final weight function mapping F to $\mathbb{K}$.

Given a transition $e \in E$, we denote by $i[e]$ its (input) label, $w[e]$ its weight, $p[e]$ its origin (or previous state) and $n[e]$ its destination state (or next state). Given a state $q \in Q$, we denote by $E[q]$ the set of transitions leaving q, and by $E^R[q]$ the set of transitions entering q.

A *path* $\pi = e_1 \cdots e_k$ in A is an element of E^* with consecutive transitions: $n[e_{i-1}] = p[e_i]$, $i = 2, \ldots, k$. We extend n, and p to paths by setting: $n[\pi] = n[e_k]$, and $p[\pi] = p[e_1]$. We denote by $P(q, q')$ the set of paths from q to q'. P can be extended to subsets $R \subseteq Q$ $R' \subseteq Q$, by:

$$P(R, R') = \bigcup_{q \in R, q' \in R'} P(q, q')$$

The labeling function i and the weight function w can also be extended to paths by defining the label of a path as the concatenation of the labels of its constituent transitions, and the weight of a path as the $\otimes$-product of the weights of its constituent transitions:

$$\begin{aligned} i[\pi] &= i[e_1] \cdots i[e_k] \\ w[\pi] &= w[e_1] \otimes \cdots \otimes w[e_k] \end{aligned}$$

Given a string $x \in \Sigma^*$, we denote by $\Pi(x)$ the set of paths from I to F labeled with x:

$$\Pi(x) = \{\pi \in P(I, F) : i[\pi] = x\}$$

The output weight associated by A to an input string $x \in \Sigma^*$ is:

$$A \cdot x = \bigoplus_{\pi \in \Pi(x)} \lambda(p[\pi]) \otimes w[\pi] \otimes \rho(n[\pi])$$

If $\Pi(x) = \emptyset$, $A \cdot x$ is defined to be $\overline{0}$.

These definitions can be easily generalized to include the case of weighted automata with ϵ-transitions.

3 Language recognition with weighted automata

3.1 Definition

The definition of language recognition with unweighted automata is classical. A string x is said to be *recognized* or *accepted* by A if there exists a path from an initial state to a final state labeled with x: $\Pi(x) \neq \emptyset$. There exists a simple algorithm for testing for the emptiness of $\Pi(x)$ (Aho, Hopcroft, and Ullman, 1974). It can be viewed as a special case of the intersection, or composition algorithm for automata. Indeed, the input string x can be represented by a simple linear automaton, $X(x)$. Thus, the emptiness of $\Pi(x)$ is equivalent to that of the intersection automaton $A \cap X(x)$. We introduce a definition of language recognition with weighted automata which can be viewed as a generalization of the classical recognition with unweighted automata.

Definition 2 *Let* $\mathbb{J} \subseteq \mathbb{K}$ *be a subset of* $\mathbb{K}$*. We say that a string* $x \in \Sigma^*$ *is* $\mathbb{J}$-recognized *or* $\mathbb{J}$-accepted *by the weighted automaton* A *if* $A \cdot x \in \mathbb{J}$*.*

The definition is a generalization of recognition with unweighted automata. Indeed, unweighted automata can be viewed as weighted automata over the boolean semiring $\mathcal{B}$ =

$(\{0,1\}, \vee, \wedge, 0, 1)$. Classical recognition with unweighted automata is then equivalent to $\mathbb{J}$-recognition with $\mathbb{J} = \{1\}$.

In what follows, we can assume without loss of generality that the weighted automaton A admits no ϵ-transition cycle since a general ϵ-removal algorithm can be used to construct an equivalent automaton with no ϵ-transition (Mohri, Pereira, and Riley, 1999).

3.2 Algorithm

We now present a generic recognition algorithm with weighted automata. Given a string x, recognition with a weighted automaton A can be done in two steps:

1. computation of $A \cdot x$,
2. membership test $A \cdot x \in \mathbb{J}$.

The following proposition shows that the cost of the first step of the algorithm is similar to that of the classical recognition with unweighted automata. It can be performed in time linear with respect to the length of x.

Proposition 1 *Let $A = (\Sigma, Q, I, F, E, \lambda, \rho)$ be a weighted automaton over the semiring $\mathbb{K}$. Then there exists an algorithm for computing $A \cdot x$ for any $x \in \Sigma^*$ in time $O(|A| \cdot |x| \cdot (T_\oplus + T_\otimes))$, where $T_\oplus$ represents the cost of the sum operation of the semiring, and $T_\otimes$ that of multiplication.*

Proof. As mentioned previously, x can be represented by a linear acceptor $X(x)$ in time $O(|x|)$. Using the general composition (or intersection) algorithm for weighted automata (Mohri, Pereira, and Riley, 1996), we can compute $A \circ X(x)$ in time $O(|A| \cdot |X(x)|)$, that is in $O(|A| \cdot |x|)$ since the number of transitions of $X(x)$ is equal to the length of x. By definition of composition, the successful paths in the weighted automaton $A \circ X(x)$ are all labeled with x, and the $\oplus$-sum of the weights of all these paths is exactly $A \cdot x$.

The computation of the sum of the weights of all paths from a fixed source state to all other states, or to the set of final states, can be done using an algorithm that is a generalization of the classical single-source shortest-paths algorithms to the case of directed graphs weighted over a semiring (Mohri, 1998). We denote by GSD the function computed by this generalized single-source shortest-distance algorithm. In the case of acyclic graphs, the algorithm works with any (right) semiring. Its complexity is linear in the size of the input graph, and the cost of the semiring operations. Since A admits no ϵ-transition cycle, the weighted automaton $A \circ X(x)$ is acyclic. Thus, the total cost of the algorithm for computing $A \cdot x$ is $O(|A| \cdot |x| \cdot (T_\oplus + T_\otimes))$. $\square$

Note that the algorithms used in the proof of the proposition, composition and generic single-source shortest-distance algorithm, are already used in various applications for manipulating weighted automata. Following the proof of the proposition, the procedure for recognition with weighted automata can be described by the following formula:

$$\mathrm{GSD}(A \circ X(x)) \in \mathbb{J}$$

The complexity of the second stage of the algorithm depends on the subset $\mathbb{J}$. If we assume that an equality test can be performed in constant time in the semiring $\mathbb{K}$, then membership can be tested in constant time for any finite subset $\mathbb{J}$. In particular, when $\mathbb{J}$ is reduced to a single element, the membership test can be performed in constant time.

In the following, we will be focusing on specific cases where $\mathbb{J}$ is reduced to a single element. It should be clear though that the recognition power of weighted automata can be increased by considering larger or more complex subsets. This does not necessarily imply a more costly membership test. For an example of this fact, consider the case $\mathbb{K} = \mathbb{R}$, and $\mathbb{J} = \mathbb{R}_+$, the test is reduced to: $A \cdot x \geq 0$ which can still be performed in constant time.

3.3 Extension by composition with finite-state transducers and intersection with regular languages

This section shows that once a language L has been shown to be recognizable by weighted automata over a semiring $\mathbb{K}$, then several classes of languages are also recognizable by the same devices and with the same time complexity.

Let τ be a transduction from Σ^* to Δ^* and let $Y \subseteq \Delta^*$, we define $\tau^{-1}(Y)$ by:

$$\tau^{-1}(Y) = \{x \in \Sigma^* : \tau(x) \subseteq Y\}$$

We say that a language L' is an *inverse rational image* of language L if there exists a rational transduction τ such that:

$$\tau^{-1}(L) = L'$$

When further L is a rational inverse of L', L and L' are said to be *equivalent.* [1]

Recognition can be extended to other languages by composition with finite-state transducers and intersection with finite automata.

Proposition 2 *Let $L \subseteq \Sigma^*$ be a language $\mathbb{J}$-recognizable with a weighted automaton $A = (\Sigma, Q, I, F, E, \lambda, \rho)$ over the semiring $\mathbb{K}$.*

1. *If $L' \subseteq \Delta^*$ is an inverse rational image of $L \subseteq \Sigma^*$, then L' is $\mathbb{J}$-recognizable;*
2. *Let R a be regular language, then there exists a weighted automaton A' $\mathbb{J}$-recognizing $L' = L \cap R$.*

Proof. Let τ be a rational transduction such that $\tau^{-1}(L) = L'$, and let T be a finite-state transducer realizing τ. Then if $x' \in L'$, $\tau(x) \subseteq L$. Conversely, let $x' \in \Delta^*$ and assume that $\tau(x) \subseteq L$, then $\tau^{-1}(\tau(x)) \subseteq L'$ which implies $x' \in L'$.

Thus $x' \in L'$ iff $\tau(x) \subseteq L$, that is iff $\mathrm{GSD}(A \circ \pi_1(T \circ X(x'))) \in \mathbb{J}$ where $\pi_1(T')$ is the automaton obtained by projection of the transducer T' over the input labels. This is equivalent to: $\mathrm{GSD}(A \circ T \circ X(x')) \in \mathbb{J}$ since input or output labels do not affect the single-source shortest-distance algorithm. Hence it is also equivalent to: $\mathrm{GSD}(\pi_2(A \circ T) \circ X(x')) \in \mathbb{J}$ where π_2 is the output projection for transducers. Thus, $A' = \pi_2(A \circ T)$ is a weighted automaton $\mathbb{J}$-recognizing L'. This proves the first part of the proposition.

Let R be a regular language and let A' be a finite automaton accepting R. Then, clearly, $x \in L' = L \cap R$ iff $x \in R$ and $\mathrm{GSD}(A \circ X(x)) \in \mathbb{J}$. Thus, $x \in L'$ iff $\mathrm{GSD}((A \cap A') \circ X(x)) \in \mathbb{J}$. $A \cap A'$ (or $A \circ A'$) $\mathbb{J}$-recognizes L'. □

Recall that the *cylinder generated by* L is the set of languages L' such that $L' = \phi^{-1}(L)$ with ϕ a morphism or $L' = L \cap R$ with R regular (Berstel, 1979).

[1] Note that this notion of rational equivalence does not coincide with the classical notion of (rationally) equivalent languages described by (Berstel, 1979).

Proposition 3 *Let $L \subseteq \Sigma^*$ be a language $\mathbb{J}$-recognizable with a weighted automaton $A = (\Sigma, Q, I, F, E, \lambda, \rho)$ over the semiring $\mathbb{K}$. Then:*

1. *Any language equivalent to L is $\mathbb{J}$-recognizable;*
2. *Let $L' \subseteq \Sigma^*$ and ϕ be a morphism such that $L' = \phi^{-1}(L)$, then L' is $\mathbb{J}$-recognizable;*
3. *Any language L' that belongs to the cylinder generated by L is $\mathbb{J}$-recognizable.*

Proof. The first two statements are direct consequences of proposition 2. By definition, if L' is a language equivalent to L, then L' is an inverse rational image of L. Also, any morphism ϕ is a rational transduction and $x' \in \phi^{-1}(L)$ iff $\phi(x') \in L$.

The result just proved combined with proposition 2, shows that the cylinder generated by L is also $\mathbb{J}$-recognizable. □

The proposition shows in particular that if L can be recognized with weighted automata, then any language in the cylinder generated by L can be recognized with the same running time complexity. This result can be easily generalized: cylinders preserve recognition complexity for any recognition algorithm used (Berstel, 1979).

We will use proposition 3 in the following to extend our recognition results for a language L to the class of languages equivalent to L or to the cylinder generated by L.

Note that the rational image of a $\mathbb{J}$-recognizable language can also be recognized using the same algorithms, but the corresponding procedure does not preserve the complexity of recognition of L. Recall that a language L' is said to be a *rational image* of L if there exists a rational transduction τ such that $\tau(L) = L'$. If L is $\mathbb{J}$-recognized with a weighted automaton A then L' can be recognized using the following property: $x \in L'$ iff $\tau^{-1}(x') \cap L \neq \emptyset$. Let T be the inverse of a transducer realizing τ, then $x \in L'$ iff there exists a string x such that $\pi_1(A \circ T \circ X(x')) \cdot x \in \mathbb{J}$.

4 Semiring of real numbers

In this section we show that the class of context-free languages equivalent to the language of palindromes or to the symmetric language of

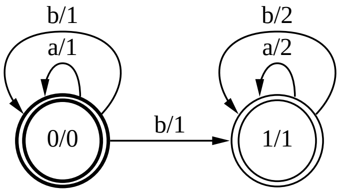

Figure 1: Weighted automaton computing the integer value of binary numbers, $\Sigma = \{a, b\}$, $a = 0$, $b = 1$. Initial states are represented by bold circles, final states by double circles. Inside each circle, the first number indicates the state number, the second, at final states only, the value of the final weight function ρ at that state.

second order can be efficiently recognized with weighted automata over the semiring of real numbers $\mathcal{R} = (\mathbb{R}, +, \cdot, 0, 1)$.

Let $k = |\Sigma|$ be the size of the alphabet, we can order the elements of Σ in an arbitrary way, and write: $\Sigma = \{0, \ldots, k-1\}$ by identifying each element a_k of the alphabet with its rank k. A string $x = x_0 \cdots x_n$ can thus be considered as a number in base k. We denote by $\langle x \rangle$ its integer value:

$$\langle x \rangle = \sum_{i=0}^{n} x_i k^{n-i}$$

Our construction of a weighted automaton recognizing palindromes is based on that of one that computes the integer value of an input string.

Proposition 4 *There exists a weighted automaton $A = (\Sigma, Q, I, F, E, \lambda, \rho)$ over the semiring of real numbers such that for any string $x \in \Sigma^*$, $A \cdot x = \langle x \rangle$.*

Proof. Indeed, consider the power series S defined by:

$$S = \sum_{i=1}^{k-1} (\Sigma)^* (ia_i)(k\Sigma)^*$$

S is clearly a rational power series as a sum of products of the rational power series: Σ^*, ia_i, and $(k\Sigma)^*$. Thus S can be realized by a weighted automaton A (Schützenberger, 1961). Figure 1 shows that weighted automaton for the case $k = 2$.

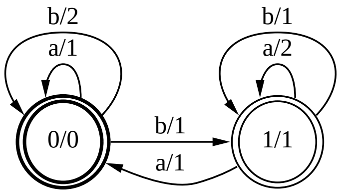

Figure 2: Other example of a weighted automaton computing the integer value of binary numbers, $\Sigma = \{a, b\}$, $a = 0$, $b = 1$.

Let $x = x_0 \ldots x_n$ be a string. Since A realizes S, we have: $A \cdot x = (S, x)$. Adding 0 to S doesn't change its definition, thus we can write:

$$S = \sum_{i=0}^{k-1} (\Sigma)^* (ia_i)(k\Sigma)^*$$

By definition of the product of power series:

$$(S, x) = \sum_{i=0}^{k-1} \sum_{ua_i v = x} (\Sigma, u)(ia_i, a_i)(k\Sigma, v)$$

where the second sum runs over all possible decompositions of the string x into a prefix u followed by a_i followed by a suffix v. Thus:

$$\begin{aligned}(S, x) &= \sum_{i=0}^{k-1} \sum_{ua_i v = x} 1^{|u|} i k^{|v|} \\ &= \sum_{i=0}^{k-1} \sum_{x_j = a_i} i k^{(n-j)} = \sum_{j=0}^{n} x_j k^{(n-j)}\end{aligned}$$

This proves the proposition. □

Note that there are weighted automata other than the one described in the proof of the proposition which have the same property. Figure 2 shows another weighted automaton that also computes integer values of input strings in base 2.

Let x be a string over the alphabet Σ, and denote by x^R its mirror image. Note that in the computation of the integer value $\langle x \rangle$ of x, leading 0's are ignored. However, if two strings x and x' have the same integer values $\langle x \rangle = \langle x' \rangle$ and have the same lengths $|x| = |x'|$, then they are necessarily equal. Thus, since a string and

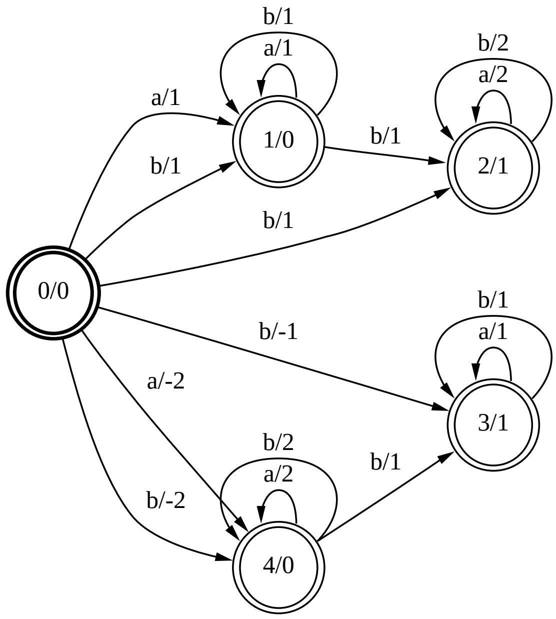

Figure 3: Weighted automaton recognizing palindromes in the semiring of real numbers, $\Sigma = \{a, b\}$.

its mirror image have the same length, x is a palindrome iff $\langle x \rangle = \langle x^R \rangle$. This is the characterization that we use to construct a weighted automaton recognizing palindromes.

Theorem 1 *Let Σ be a finite alphabet. Then the class of languages equivalent to the language of palindromes and the cylinder generated by the language of palindromes can be* 0*-recognized by weighted automata over the semiring of real numbers.*

Proof. By proposition 4, there exists a weighted automaton B computing the integer values of strings defined over the alphabet Σ. We denote by B^R the reverse automaton of B. B^R is obtained from B by reversing the direction of the transitions, exchanging initial states and final states, and exchanging the initial weight function and the final weight function. Now define the weighted automaton A by:

$$A = B - B^R$$

A can be constructed by taking the sum (or union) of B and $-B^R$. By construction, for any string x, $A \cdot x = \langle x \rangle - \langle x^R \rangle$. Hence, $A \cdot x = 0$ iff x is a palindrome. Thus, the language of palindromes can be 0-recognized by a weighted automaton over the semiring of real numbers. By proposition 3, this result can be extended to the class of languages equivalent to the language of palindromes, and to the cylinder generated by this language. This ends the proof of the theorem. □

Using the recognition algorithm presented in the previous section, A can recognize palindromes in time linear in the length of the input string. Figure 3 shows the weighted automaton A constructed in the proof of the theorem for the case $\Sigma = \{a, b\}$. A similar result holds for the symmetric language of second order S_2. Let $\Sigma = \{a, b, \overline{a}, \overline{b}\}$. For any string $x = x_{i_1} \cdots x_{i_p} \in \{a, b\}^*$, we denote by $\overline{x}$ the string defined by: $\overline{x} = \overline{x}_{i_n} \cdots \overline{x}_{i_p}$. Recall that S_2 is defined by:

$$S_2 = \{x\overline{x} : x \in \{a, b\}^*\}$$

S_2 is a generator of the cone of linear languages Lin.

Theorem 2 *Let Σ be a finite alphabet. Then the class of languages equivalent to the symmetric language of second order S_2 and the cylinder generated by S_2 can be* 0*-recognized by weighted automata over the semiring of real numbers.*

Proof. We introduce two new symbols a_0 and $\overline{a}_0$, By proposition 4, there exists a weighted

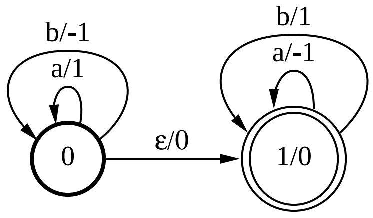

Figure 4: Weighted automaton over the real tropical semiring recognizing $D_1'^*$.

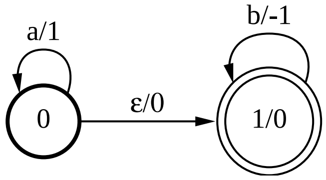

Figure 5: Weighted automaton over the real tropical semiring 0-recognizing the language $S_1 = \{a^n b^n : n \in \mathbb{N}\}$.

automaton $B(a_0, a, b)$ computing the integer values of strings defined over the alphabet $\{a_0, a, b\}$. Strings over this alphabet can thus be considered as numbers in base 3 with a_0 corresponding to 0. Let $\overline{C}$ be a weighted automaton realizing $\{\overline{a}, \overline{b}\}^*$. $\overline{C}$ associates 1 to any string over the alphabet $\{\overline{a}, \overline{b}\}^*$.

Similarly, by proposition 4, there exists a weighted automaton $B(\overline{a}_0, \overline{a}, \overline{b})$ computing integer values of strings defined over the alphabet $\{\overline{a}_0, \overline{a}, \overline{b}\}$, and we define C as the weighted automaton realizing $\{a, b\}^*$. Then the automaton A defined by:

$$A = B(a_0, a, b) \cdot \overline{C} - (B(\overline{a}_0, \overline{a}, \overline{b}) \cdot C)^R$$

0-recognizes S_2. Indeed, by definition of $B(a_0, a, b)$, $B(a_0, a, b)\overline{C}$ associates to a string over Σ its integer value by ignoring potential final symbols in $\{\overline{a}, \overline{b}\}^*$. Similarly, $(B(\overline{a}_0, \overline{a}, \overline{b})C)^R$, associates to the reverse of a string over Σ its integer value by ignoring potential initial symbols over $\{a, b\}^*$. Clearly, by definition of S_2, a string belongs to S_2 iff these two integer values are equal (note that there is no leading 0 in the computation of these integers since a_0 and $\overline{a}_0$ are not in Σ). By proposition 3, this result extends to the cylinder generated by S_2 as well as to the class of languages equivalent to S_2. □

5 Real tropical semiring

Weighted automata over the tropical semiring $\mathcal{T} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$ are used in many text and speech processing applications (Mohri, 1997). The weights are often interpreted as negative log of probabilities, thus they are added along the paths, and given an input string x, the corresponding output weight is the minimum of the weights of all the paths labeled with x since the Viterbi approximation is used.

This section shows that the same weighted automata currently used in various applications can be used to recognize efficiently a class of context-free languages that includes the Dyck language of first order (Berstel, 1979). Note that the composition algorithm used for language recognition with these weighted automata is exactly the one used in speech processing applications, and that the generic shortest-distance algorithm coincides with the classical single-source shortest-paths algorithm here.

We first give a combinatorial characterization of the strings of the Dyck language of first order $D_1'^*$. This will be used to construct directly a weighted automaton recognizing $D_1'^*$.

Recall that the Dyck language of first order is the set of strings with well-formed parentheses. We denote by a the left parenthesis and by b the right parenthesis. Thus, $aabb$ and $aababb$ belong to $D_1'^*$, while $abba$ or $aabbba$ do not. Given a string x over the alphabet $\Sigma = \{a, b\}$, we denote by $|x|_a$ the number of a's in x and by $|x|_b$ the number of b's, and we define $\|x\|$ by:

$$\|x\| = |x|_a - |x|_b$$

Given a string u, we write $u \leq_p x$ when u is a prefix of x.

Lemma 1 *Let x be a string over the alphabet $\Sigma = \{a, b\}$. Then x belongs to $D_1'^*$ iff:*

$$\min_{u_1 u_2 = x} (\|u_1\| - \|u_2\|) = 0$$

Proof. We will use for the proof a classical property of $D_1'^*$ (Berstel, 1979): $x \in D_1'^*$, iff $\|x\| = 0$ and $\|u\| \geq 0$ for any prefix u of x.

Note that for any decomposition of x into a prefix u_1 and suffix u_2, $\|x\| = \|u_1\| + \|u_2\|$. Thus $\|u_1\| - \|u_2\| = 2\|u_1\| - \|x\|$.

Assume first that $x \in D_1'^*$, then $\|x\| = 0$ and $\|u_1\| \geq 0$. Thus: $\|u_1\| - \|u_2\| = 2\|u_1\| \geq 0$, and $\|u_1\| - \|u_2\| = \|x\| = 0$ for $u_1 = w$ and $u_2 = \epsilon$.

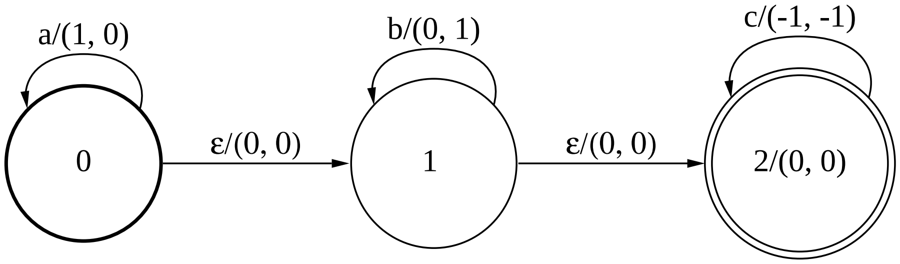

Figure 6: Weighted automaton over the $\mathbb{R}^2$-tropical semiring $(0,0)$-recognizing the language $\{a^n b^n c^n : n \in \mathbb{N}\}$.

Conversely, assume that $\min_{u_1 u_2 = x}(\|u_1\| - \|u_2\|) = 0$. In particular, ($u_1 = \epsilon$, $u_2 = x$): $-\|x\| \geq 0$, and ($u_1 = x$, $u_2 = \epsilon$): $\|x\| \geq 0$, hence $\|x\| = 0$. Thus for any decomposition $u_1 u_2 = x$,

$$2\|u_1\| = \|u_1\| - \|u_2\|$$

Hence $\|u_1\| \geq 0$ for any prefix u_1 of x. This ends the proof of the lemma. □

We use the lemma to construct a weighted automaton 0-recognizing $D_1'^*$.

Theorem 3 *Let $\Sigma = \{a, b\}$. Then the class of languages equivalent to the language of well-formed parentheses $D_1'^*$ over the alphabet Σ and the cylinder generated by $D_1'^*$ can be 0-recognized by weighted automata over the real tropical semiring.*

Proof. Consider the power series S defined by:

$$S = (a - b)^*(b - a)^*$$

S is clearly rational as a product of two rational power series. Thus, it can be realized by a weighted automaton. Figure 4 shows a weighted automaton A representing S. By definition of the product of power series in the tropical semiring, for any string x:

$$A \cdot x = (S, x) = \min_{u_1 u_2 = x}((a-b)^*, u_1) + ((b-a)^*, u_2)$$

Thus: $A \cdot x = (S, x) = \min_{u_1 u_2 = x}(\|u_1\| - \|u_2\|)$. By proposition 3, the result can be extended to the languages equivalent to $D_1'^*$ and to the cylinder generated by $D_1'^*$. This ends the proof of the theorem. □

Thus the weighted automaton of figure 4 can be used to recognize $D_1'^*$ in linear time. There are other classes of languages that can be recognized similarly by weighted automata over the tropical semiring. A simpler example is that of the context-free language $S_1 = \{a^n b^n : n \in \mathbb{N}\}$. Clearly, it is 0-recognized by the weighted automaton of figure 5.

6 Conclusion

A new definition of language recognition with weighted automata was given. A generic linear time algorithm for language recognition with weighted automata was also presented: the algorithm is generic in the sense that it works with any right semiring. Several classes of context-free languages (the class of languages equivalent to the language of palindromes, the class of languages equivalent to S_2, the class of languages equivalent to $D_1'^*$, and the class of languages equivalent to S_1) were proved to be recognizable in linear time with this algorithm.

Weighted automata can be used to recognize languages of higher order. Since the cross product of two semirings is a semiring, weighted automata over cross products can also be used.

An example that illustrates both these points is the recognition of $S_3 = \{a^n b^n c^n : n \in \mathbb{N}\}$. The cross product of the real-tropical semiring by itself, is a semiring called the $\mathbb{R}^2$-tropical semiring. One can easily construct a weighted automaton over the $\mathbb{R}^2$-tropical semiring $(0,0)$-recognizing S_3. Figure 6 shows that weighted automaton. It can be used to recognize that language in time linear in the size of the input string using the generic recognition algorithm presented in previous sections. S_3 can also be directly recognized by weighted automata over the semiring of real numbers using prime numbers, or by weighted automata over the tropical semiring using rational numbers.

References

Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman. 1974. *The design and analysis of computer algorithms.* Addison Wesley: Reading, MA.

Berstel, Jean. 1979. *Transductions and Context-Free Languages.* Teubner Studienbucher: Stuttgart.

Berstel, Jean and Christophe Reutenauer. 1988. *Rational Series and Their Languages.* Springer-Verlag: Berlin-New York.

Kleene, S. C. 1956. Representation of events in nerve nets and finite automata. *Automata Studies.*

Mohri, Mehryar. 1998. General algebraic frameworks and algorithms for shortest-distance problems. Technical Memorandum 981210-10TM, AT&T Labs - Research, 62 pages.

Mohri, Mehryar. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:2.

Mohri, Mehryar, Fernando C. N. Pereira, and Michael Riley. 1996. Weighted automata in text and speech processing. In *ECAI-96 Workshop, Budapest, Hungary.* ECAI.

Mohri, Mehryar, Fernando C. N. Pereira, and Michael Riley. 1999. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, to appear.

Eilenberg, Samuel. 1974. *Automata, Languages and Machines*, volume A-B. Academic Press.

Schützenberger, Marcel Paul. 1961. On the definition of a family of automata. *Information and Control*, 4.