A disambiguation algorithm for weighted automata

Mehryar Mohri [a,b], Michael D. Riley [b,*]

[a] *Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012, United States*
[b] *Google Research, 76 Ninth Avenue, New York, NY 10011, United States*

A B S T R A C T

We present a disambiguation algorithm for weighted automata. The algorithm admits two main stages: a pre-disambiguation stage followed by a transition removal stage. We give a detailed description of the algorithm and the proof of its correctness. The algorithm is not applicable to all weighted automata but we prove sufficient conditions for its applicability in the case of the tropical semiring by introducing the *weak twins property*. In particular, the algorithm can be used with any weighted automaton over the tropical semiring for which the weighted determinization algorithm terminates and with any acyclic weighted automaton over an arbitrary weakly left divisible cancellative and commutative semiring. While disambiguation can sometimes be achieved using weighted determinization, our disambiguation algorithm in some cases can return a result that is exponentially smaller than any equivalent deterministic automaton. We also present some empirical evidence of the space benefits of disambiguation over determinization in speech recognition and machine translation applications.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Weighted finite automata and transducers are widely used in applications. Most modern speech recognition systems used for hand-held devices or spoken-dialog applications use weighted automata and their corresponding algorithms for the representation of their models and their efficient combination and search [19,2]. Similarly, weighted automata are commonly used for a variety of tasks in machine translation [10] and other natural language processing applications [11], computational biology [7], image processing [1], optical character recognition [5], and many other areas.

A problem that arises in several applications is that of *disambiguation of weighted automata*: given an input weighted automaton, the problem consists of computing an equivalent weighted automaton that is *unambiguous*, that is one with no two accepting paths labeled with the same string. Fig. 1 shows two equivalent weighted automata, one ambiguous and one not.

The need for disambiguation is often motivated by the common problem of determining the most probable string, or more generally the *n* most likely strings of a *lattice*, that is an acyclic weighted automaton generated by a complex model, such as those used in machine translation, speech recognition, information extraction, and many other natural language processing and computational biology systems. A lattice compactly represents the model's most likely hypotheses. It defines a probability distribution over the strings and is used as follows: the weight of an accepting path is obtained by multiplying the weights of its component transitions and the weight of a string obtained by summing up the weights of accepting paths

---

* Corresponding author.
  E-mail addresses: mohri@cims.nyu.edu (M. Mohri), riley@google.com (M.D. Riley).
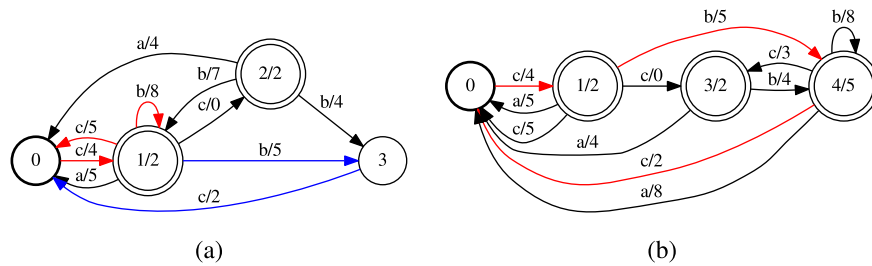
Fig. 1. Example of (a) an ambiguous weighted automaton and (b) an equivalent, unambiguous automaton. In figures here, initial states are depicted by a bold circle (always with initial weight $\overline{1}$) and final states by double circles containing their final weight. Transitions are labeled with their symbol and weight. In this example, weights are over the tropical semiring. The string cbcc labels two accepting paths, colored as cbcc and cbcc, in (a) but only one path, cbcc, in (b). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

labeled with that string. In general, there may be many accepting paths labeled with a given string. Clearly, if the lattice were unambiguous, a standard shortest-paths or n-shortest-paths algorithm [9] could be used to efficiently determine the n most likely strings. When the lattice is not unambiguous, the problem is more complex and can be solved using weighted determinization [20]. An alternative solution, which we will show has benefits, consists of first finding an unambiguous weighted automaton equivalent to the lattice and then running an n-shortest-paths algorithm on the resulting weighted automaton. A similar need for disambiguation appears when computing the marginals of a given weighted transducer.

Another common problem where disambiguation is needed is that of sampling strings from a weighted automaton according to the probability distribution it induces. This weighted automaton may be defined over a semiring different from the probability semiring but with the same weight set and the same multiplicative operation. This problem arises, for example, in the context of on-line learning with path experts [6]. Sampling from that weighted automaton directly is a difficult problem. But, if instead an equivalent unambiguous weighted automaton can be computed, then the additive operation of the semiring would be inconsequential for that weighted automaton. One can then equivalently work in the probability semiring and use a straightforward sampling method.

In general, one way to determine an equivalent unambiguous weighted automaton is to use the weighted determinization algorithm [17]. This, however, admits several drawbacks. First, weighted determinization cannot be applied to all weighted automata. This is both because not all weighted automata admit an equivalent deterministic weighted automaton but also because even for some that do, the weighted determinization algorithm may not halt. Sufficient conditions for the application of the algorithm have been given [17,3]. In particular the algorithm can be applied to all acyclic weighted automata over an arbitrary semiring and to all weighted automata over the tropical semiring admitting the twins property. Nevertheless, a second issue is that in some cases where weighted determinization can be used, the size of the resulting deterministic automaton is prohibitively large.

This paper presents a new disambiguation algorithm for weighted automata extending to the weighted case the algorithm of [18] – the weighted case is significantly more complex and this extension non-trivial. As we shall see, our disambiguation algorithm applies to a broader family of weighted automata than determinization in the tropical semiring: we show that if a weighted automaton can be determinized using the algorithm of [17], then it can also be disambiguated using the algorithm presented in this paper (see Section 6). Furthermore, for some weighted automata, the size of the unambiguous weighted automaton returned by our algorithm is exponentially smaller than that of any equivalent deterministic weighted automata. In particular, our algorithm leaves the input unchanged if it is unambiguous, while the size of the automaton returned by determinization for some unambiguous weighted automata is exponentially larger. An example is given in Section 7. We also present empirical evidence showing the benefits of weighted disambiguation over determinization in applications. Our algorithm applies in particular to unweighted finite automata. Note that it is known that for some non-deterministic finite automata of size n the size of an equivalent unambiguous automaton is at least $\Omega(2^{\sqrt{n}})$ [22], which gives a lower bound on the time and space complexity of any disambiguation algorithm for finite automata.

Our disambiguation algorithm for weighted automata is presented in a general way and for a broad class of semirings. Nevertheless, the algorithm is limited in several ways. First, not all weighted automata admit an equivalent unambiguous weighted automaton. But, even for some that do, our algorithm may not succeed. The situation is thus similar to that of weighted determinization. However, we present sufficient conditions based on a new notion of *weak twins property* under which our algorithm can be used. In particular, our algorithm applies to all acyclic weighted automata and more generally to all weighted automata for which the weighted determinization algorithm of [17] terminates. Our algorithm admits two stages. The first stage called *pre-disambiguation* constructs a weighted automaton with several key properties, including the property that paths leaving the initial state and labeled with the same string have the same weight. The second stage consists of removing some transitions to make the result unambiguous. Our disambiguation algorithm can be applied whenever pre-disambiguation terminates.

The paper is organized as follows. In Section 2, we review previous work on this and related problems. In Section 3, we introduce some preliminary definitions and notation relevant to the description of our algorithm. Section 4 describes our pre-disambiguation algorithm and proves some key properties of its result. We describe in fact a family of pre-

disambiguation algorithms parameterized by a relation R over the set of pairs of states. A simple instance of that relation is for two states to be equivalent when they admit a path labeled by the same string leading to a final state. In Section 5, we describe the second stage, which consists of transition removal, and prove the correctness of our disambiguation algorithm. In Section 6, we introduce the notion of *weak twins property* which we use to prove the sufficient conditions for the application of pre-disambiguation and thus the full disambiguation algorithm. The proofs under this condition are given in the case of weighted automata for the tropical semiring. For more general weakly left divisible, cancellative and commutative semirings, proofs are given for the application of disambiguation when the semiring is finite or the automaton is acyclic. Finally, in Section 7, we present experiments that compare weighted disambiguation to determinization in speech recognition and machine translation applications. Our implementation of these algorithms used in these experiments is available through a freely available OpenFst library [4]. Detailed proofs for several of our results are reserved for the appendix.

2. Previous work

We refer to [18] for an extensive discussion of disambiguation algorithms for unweighted automata and finite-state transducers, in particular the algorithm of Schützenberger. As already mentioned, our weighted disambiguation algorithm is an extension of the unweighted disambiguation of [18]. In fact, it coincides with that algorithm in the special case of the Boolean semiring.

In the weighted case, we already mentioned and discussed weighted determinization [17] as a possible disambiguation algorithm in some cases. There are two other disambiguation procedures described in the literature: one for the special case of finitely ambiguous min-plus automata [15], another one for the special case of polynomially ambiguous min-plus weighted automata [14].

The procedure described by [15] for the disambiguation of a finitely ambiguous min-plus automaton $\mathcal{A}$ (when it exists) consists of first decomposing $\mathcal{A}$ into a semiring-sum of unambiguous weighted automata. This is done by determinizing an unweighted version of $\mathcal{A}$, which results in $\mathcal{D}$, and using the intersection of $\mathcal{D}$ with $\mathcal{A}$. This step is similar to the algorithm of Schützenberger for the disambiguation of finite-state transducers. Two states of $\mathcal{A}$ paired with the same state of $\mathcal{D}$ in the intersection are by definition reachable by the same string. Thus, ambiguities appear when such two states admit a transition with the same label to the same state or if they are both final. To avoid such ambiguities, two distinct weighted automata are created, each with a copy of one of such transitions and the states and transitions that can be read from its destination. The computational details, including data structures or the details of some computational steps are not presented by the authors. The computational complexity of the algorithm is also not discussed or mentioned by the authors. In general, it seems that the number of unambiguous weighted automata thereby created can be exponential in the number of ambiguities of $\mathcal{A}$. Furthermore, the size of each of these unambiguous automata can be in the order of the size of $\mathcal{D} \cap \mathcal{A}$, which can be exponential in the size of $\mathcal{A}$. The second step of the algorithm consists of using the weighted automaton $\mathcal{P}$ that is the cross-product of all these unambiguous automata to construct an unambiguous weighted automaton equivalent to $\mathcal{A}$ using the *victorious coordinates* of strongly connected components or simple circuits. It seems that the computational cost of the construction of the resulting unambiguous automaton can be exponential in the size of $\mathcal{P}$, in which case the overall complexity of the algorithm is super-exponential. This is not completely clear and the authors do not provide a detailed description. But, one advantage of the procedure of [15] is that, as part of the same procedure, it can first test the existence of an equivalent unambiguous weighted automaton for $\mathcal{A}$. This procedure is quite different from our algorithm. It does not benefit from the relation we define over pairs of $\mathcal{A}$ to determine if two states admit the same future. It does not extend to weighted automata with an infinite number of ambiguities since $\mathcal{A}$ must then be decomposed into infinitely many unambiguous automata. It does not apply to different semirings and seems to admit a super-exponential complexity. Our algorithm seems simpler and benefits from a more favorable computational complexity. For unambiguous inputs, the complexity of our algorithm is only quadratic (time to compute for all pairs if they admit a common future) and the output weighted automata coincide with the input ones. In contrast, because of the determinization step, the worst case complexity of the algorithm of [15] is exponential for unambiguous inputs.

An alternative procedure was also described by [14] [pp. 598–599] for constructing an unambiguous weighted automaton (when it exists) in the specific case of polynomially ambiguous min-plus weighted automata. The construction is rather intricate and further relies on the prior determination of a threshold value Y. The authors do not give an explicit algorithm for computing Y but state that it can be inferred from [14] [Proposition 5.1]. However, the corresponding procedure seems intractable. In fact, as indicated by the authors, the cost of determining Y using that property is super-exponential. The authors of [14] do not give the running-time complexity of their procedure and do not detail various aspects, which makes a comparison difficult. But, our algorithm is much simpler and seems to be significantly more efficient. Our algorithm is also more general since it applies in particular to weighted automata over the tropical semirings that verify the weak twins property and that may be exponentially ambiguous. It is also given for a broader family of semirings. While we are not presenting guarantees for its applicability for semirings different from the tropical semiring, its applicability for at least acyclic weighted automata for those semirings is clear. One advantage of the procedures described by [14] is that the existence of an unambiguous weighted automaton is first tested, though that test procedure appears also to be very costly.

Finally, let us mention that an algorithm of Eilenberg [8] bears the same name, disambiguation, but it is in fact designed for an entirely different problem.

3. Preliminaries

Given an alphabet Σ, we will denote by $|x|$ the length of a string $x \in \Sigma^*$ and by ϵ the *empty string* for which $|\epsilon| = 0$.

The weighted automata we consider are defined over a broad class of *semirings*. A semiring is a system $(\mathbb{S}, \oplus, \otimes, \overline{0}, \overline{1})$ where $(\mathbb{S}, \oplus, \overline{0})$ is a commutative monoid with $\overline{0}$ as the identity element for $\oplus$, $(\mathbb{S}, \otimes, \overline{1})$ is a monoid with $\overline{1}$ as the identity element for $\otimes$, $\otimes$ distributes over $\oplus$, and $\overline{0}$ is an annihilator for $\otimes$.

A semiring is said to be *commutative* when $\otimes$ is commutative. Some familiar examples of (commutative) semirings are the tropical semiring $(\mathbb{R}_+ \cup \{+\infty\}, \min, +, +\infty, 0)$, the semiring of non-negative integers $(\mathbb{N}, +, \times, 0, 1)$ and of non-negative reals $(\mathbb{R}_+, +, \times, 0, 1)$. The multiplicative operation of a semiring $(\mathbb{S}, \oplus, \otimes, \overline{0}, \overline{1})$ is said to be *cancellative* if for any x, x' and z in $\mathbb{S}$ with $z \neq \overline{0}$, $x \otimes z = x' \otimes z$ implies $x = x'$. When that property holds, the semiring $(\mathbb{S}, \oplus, \otimes, \overline{0}, \overline{1})$ is also said to be *cancellative*.

A semiring $(\mathbb{S}, \oplus, \otimes, \overline{0}, \overline{1})$ is said to be *left divisible* if any element $x \in \mathbb{S} - \{\overline{0}\}$ admits a left inverse $x' \in \mathbb{S}$, that is $x' \otimes x = \overline{1}$. $(\mathbb{S}, \oplus, \otimes, \overline{0}, \overline{1})$ is said to be *weakly left divisible* if for any x and x' in $\mathbb{S}$ such that $x \oplus x' \neq \overline{0}$, there exists at least one z such that $x = (x \oplus x') \otimes z$. When the $\otimes$ operation is cancellative, z is unique and we can then write: $z = (x \oplus x')^{-1} \otimes x$.

Weighted finite automata (WFAs) are automata in which the transitions are labeled with weights, which are elements of a semiring [16]. A WFA $\mathcal{A}$ over $\mathbb{S}$ is a 7-tuple $(\Sigma, Q_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A}, E_\mathcal{A}, \lambda_\mathcal{A}, \rho_\mathcal{A})$ where: Σ is the finite alphabet of the automaton, $Q_\mathcal{A}$ is a finite set of states, $I_\mathcal{A} \subseteq Q_\mathcal{A}$ the set of initial states, $F_\mathcal{A} \subseteq Q_\mathcal{A}$ the set of final states, $E_\mathcal{A}$ a finite set of transitions which are elements of $Q_\mathcal{A} \times \Sigma \times \mathbb{S} \times Q_\mathcal{A}$, $\lambda_\mathcal{A}: I_\mathcal{A} \to \mathbb{S}$ an initial weight function, and $\rho_\mathcal{A}: F_\mathcal{A} \to \mathbb{S}$ the final weight function mapping $F_\mathcal{A}$ to $\mathbb{S}$. We may omit the $\mathcal{A}$ subscript in the above when there is no confusion in the automaton being referenced.

A path π of a WFA is an element of E^* with consecutive transitions. The label of a path is an element of Σ^* formed by concatenating its transition labels. We denote by $\text{orig}[\pi]$ the origin state and by $\text{dest}[\pi]$ the destination state of the path. A path is said to be *accepting* or *successful* when $\text{orig}[\pi] \in I$ and $\text{dest}[\pi] \in F$.

We denote by $w_\mathcal{A}[e]$ the weight of a transition e in $\mathcal{A}$ and similarly by $w_\mathcal{A}[\pi]$ the weight of path $\pi = e_1 \cdots e_n$ obtained by $\otimes$-multiplying the weights of its constituent transitions: $w_\mathcal{A}[\pi] = w_\mathcal{A}[e_1] \otimes \cdots \otimes w_\mathcal{A}[e_n]$. When $\text{orig}[\pi]$ is in I, we denote by $w_\mathcal{A}^\mathcal{I}[\pi] = \lambda_\mathcal{A}(\text{orig}[\pi]) \otimes w_\mathcal{A}[\pi]$ the weight of the path including the initial weight of the origin state. For any two subsets $U, V \subseteq Q_\mathcal{A}$ and any string $x \in \Sigma^*$, we denote by $P_\mathcal{A}(U, x, V)$ the set of paths in $\mathcal{A}$ labeled with x from a state in U to a state in V and by $W_\mathcal{A}(U, x, V)$ the $\oplus$-sum of their weights:

$$W_\mathcal{A}(U, x, V) = \bigoplus_{\pi \in P_\mathcal{A}(U,x,V)} w_\mathcal{A}[\pi].$$

When U is reduced to a singleton, $U = \{p\}$, we will simply write $W_\mathcal{A}(p, x, V)$ instead of $W_\mathcal{A}(\{p\}, x, V)$ and similarly for V. To include initial weights, we denote:

$$W_\mathcal{A}^\mathcal{I}(x, V) = \bigoplus_{\pi \in P_\mathcal{A}(I,x,V)} w_\mathcal{A}^\mathcal{I}[\pi].$$

We also denote by $\delta_\mathcal{A}(U, x)$ the set of states reached by paths starting in U and labeled with $x \in \Sigma^*$. The weight associated by $\mathcal{A}$ to a string $x \in \Sigma^*$ is defined by

$$\mathcal{A}(x) = \bigoplus_{\pi \in P_\mathcal{A}(I,x,F)} w_\mathcal{A}^\mathcal{I}[\pi] \otimes \rho_\mathcal{A}(\text{dest}[\pi]), \tag{1}$$

when $P_\mathcal{A}(I_\mathcal{A}, x, F_\mathcal{A}) \neq \emptyset$. $\mathcal{A}(x)$ is defined to be $\overline{0}$ when $P_\mathcal{A}(I_\mathcal{A}, x, F_\mathcal{A}) = \emptyset$.

A state q of a WFA $\mathcal{A}$ is said to be *accessible* if q can be reached by a path originating in $I_\mathcal{A}$. It is *coaccessible* if a final state can be reached by a path from q. Two states q and q' are *co-reachable* if they each can be reached by a path from $I_\mathcal{A}$ labeled with a common string $x \in \Sigma^*$. A WFA $\mathcal{A}$ is *trim* if all states of $\mathcal{A}$ are both accessible and coaccessible. $\mathcal{A}$ is *unambiguous* if any string $x \in \Sigma^*$ labels at most one accepting path. The intersection of two WFAs $\mathcal{A}_1$ and $\mathcal{A}_2$ over a commutative semiring is a WFA denoted by $\mathcal{A}_1 \cap \mathcal{A}_2$ that satisfies $(\mathcal{A}_1 \cap \mathcal{A}_2)(x) = \mathcal{A}_1(x) \otimes \mathcal{A}_2(x)$.

We will say that a semiring is *admissible* if it is weakly left divisible cancellative and commutative.[1] In all that follows, we will consider WFAs over an admissible semiring.

4. R-Pre-disambiguation of WFAs

We first define a relation R over pairs of states of a WFA $\mathcal{A}$. Next, we use R to define the R-*pre-disambiguation* of $\mathcal{A}$, and then analyze the properties of the result of R-pre-disambiguation.

---

[1] Our algorithms can be straightforwardly extended to the case of weakly left divisible left semirings [3].

4.1. Relation R over $Q_{\mathcal{A}} \times Q_{\mathcal{A}}$

Two states $q, q' \in Q_{\mathcal{A}}$ are said to *share a common future* if there exists a string $x \in \Sigma^*$ such that $P_{\mathcal{A}}(q, x, F)$ and $P_{\mathcal{A}}(q', x, F)$ are not empty. Let $\mathcal{M} = \mathcal{A} \cap \mathcal{A}$ denote the trim WFA obtained by intersecting $\mathcal{A}$ with itself. The states of $\mathcal{M}$ can be identified with pairs $(q, q') \in Q_{\mathcal{A}} \times Q_{\mathcal{A}}$. By definition of intersection, state (q, q') of $\mathcal{M}$ is coaccessible if and only if q and q' share a common future in $\mathcal{A}$. The set of all coaccessible states of $\mathcal{M}$ can computed in time linear in $|\mathcal{M}| \leq |\mathcal{A}|^2$. Thus, the future sharing information can be computed for all pairs of states of $\mathcal{A}$ in time $O(|\mathcal{A}|^2)$ and then checked in constant time.

Let R_f be the relation defined over $Q_{\mathcal{A}} \times Q_{\mathcal{A}}$ by $q \, R_f \, q'$ if and only if $q = q'$ or q and q' share a common future in $\mathcal{A}$. Clearly, R_f is reflexive and symmetric, but in general it is not transitive. Observe that R_f is *compatible with the inverse transition function*, that is, if $q \, R_f \, q'$, $q \in \delta_{\mathcal{A}}(p, x)$ and $q' \in \delta_{\mathcal{A}}(p', x)$ for some $x \in \Sigma^*$ with $(p, p') \in Q_{\mathcal{A}}^2$, then $p \, R_f \, p'$. We will also denote by R_0 the complete relation defined by $q \, R_0 \, q'$ for all $(q, q') \in Q_{\mathcal{A}}^2$. Clearly, R_0 is also compatible with the inverse transition function.

The construction we will define holds for any relation R out of the set of admissible relations $\mathcal{R}$ defined as the reflexive and symmetric relations over $Q_{\mathcal{A}} \times Q_{\mathcal{A}}$ that are compatible with the inverse transition function and coarser than R_f. $\mathcal{R}$ includes R_f and R_0, as well as any symmetric relation R compatible with the inverse transition function that is coarser than R_f, that is, for all $(q, q') \in Q_{\mathcal{A}}^2$, $q \, R_f \, q' \Rightarrow q \, R \, q'$. Thus, for a relation R in $\mathcal{R}$, two states q and q' that share the same future are necessarily in relation, but they may also be in relation without sharing the same future. Note in particular that R is always reflexive.

When the full WFA $\mathcal{M}$ is available, we can use that to define and compute R_f. But, in some cases only some partial information is available, for example when $\mathcal{M}$ (or its reverse) is constructed on-the-fly, starting from its final states. In such cases, an alternative relation $R \in \mathcal{R}$ may be available.

4.2. Construction

Fix a relation $R \in \mathcal{R}$. For any $x \in \Sigma^*$, and $q \in \delta_{\mathcal{A}}(U, x)$, we also denote by $\delta_{\mathcal{A}}^q(U, x)$ the set of states in $\delta_{\mathcal{A}}(U, x)$ that are in relation with q:

$$\delta_{\mathcal{A}}^q(U, x) = \delta_{\mathcal{A}}(U, x) \cap \{p : p \, R \, q\}.$$

Note that, since R is reflexive, by definition, $\delta_{\mathcal{A}}^q(I, x)$ contains q. We will assume that $W_{\mathcal{A}}^{\mathcal{I}}(x, \{p_1, \ldots, p_t\}) \neq \overline{0}$ for any $x \in \Sigma^*$ otherwise the subset corresponding to x needs not be constructed; alternatively, we can simply exclude such states from the state set. For any $x \in \Sigma^*$ and $q \in \delta_{\mathcal{A}}(I, x)$, we define the weighted subset $s(x, q)$ by

$$s(x, q) = \Big\{ (p_1, w_1), \ldots, (p_t, w_t) \colon (\{p_1, \ldots, p_t\} = \delta_{\mathcal{A}}^q(I, x))$$

$$\wedge \big(\forall i \in [1, t], w_i = W_{\mathcal{A}}^{\mathcal{I}}(x, \{p_1, \ldots, p_t\})^{-1} \otimes W_{\mathcal{A}}^{\mathcal{I}}(x, p_i) \big) \Big\}.$$

For a weighted subset $s(x, q)$, define $\mathrm{set}(s(x, q)) = \{p_1, \ldots, p_t\}$. For any WFA $\mathcal{A}$ define the WFA $\mathcal{B}$ as follows:

$$Q_{\mathcal{B}} = \{(q, s(x, q)) \colon x \in \Sigma^*, q \in \delta_{\mathcal{A}}(I_{\mathcal{A}}, x)\}$$

$$I_{\mathcal{B}} = \{(q, s(\epsilon, q)) \colon q \in I_{\mathcal{A}}\}$$

$$F_{\mathcal{B}} = \{(q, s(x, q)) \colon x \in \Sigma^*, q \in \delta_{\mathcal{A}}(I_{\mathcal{A}}, x) \cap F_{\mathcal{A}}\}$$

$$E_{\mathcal{B}} = \Big\{ ((q, s), a, w, (q', s')) \colon (q, s), (q', s') \in Q_{\mathcal{B}}, a \in \Sigma,$$

$$\exists x \in \Sigma^* \mid s = s(x, q) = \{(p_1, w_1), \ldots, (p_t, w_t)\},$$

$$s' = s(xa, q') = \{(p_1', w_1'), \ldots, (p_{t'}', w_{t'}')\},$$

$$q' \in \delta_{\mathcal{A}}(q, a), w = \bigoplus_{i=1}^{t} \big(w_i \otimes W_{\mathcal{A}}(p_i, a, \mathrm{set}(s')) \big),$$

$$\forall j \in [1, t'], w_j' = w^{-1} \otimes \Big(\bigoplus_{i=1}^{t} w_i \otimes W_{\mathcal{A}}(p_i, a, p_j') \Big) \Big\}$$

and $\forall (q, s) \in I_{\mathcal{B}}, s = \{(p_1, w_1), \ldots, (p_t, w_t)\}, \lambda_{\mathcal{B}}((q, s)) = \bigoplus_{i \in [1, t]} \lambda_{\mathcal{A}}(p_i),$

$$\forall (q, s) \in F_{\mathcal{B}}, s = \{(p_1, w_1), \ldots, (p_t, w_t)\}, \rho_{\mathcal{B}}((q, s)) = \bigoplus_{\substack{p_i \in F_{\mathcal{A}} \\ i \in [1, t]}} (w_i \otimes \rho_{\mathcal{A}}(p_i)).$$
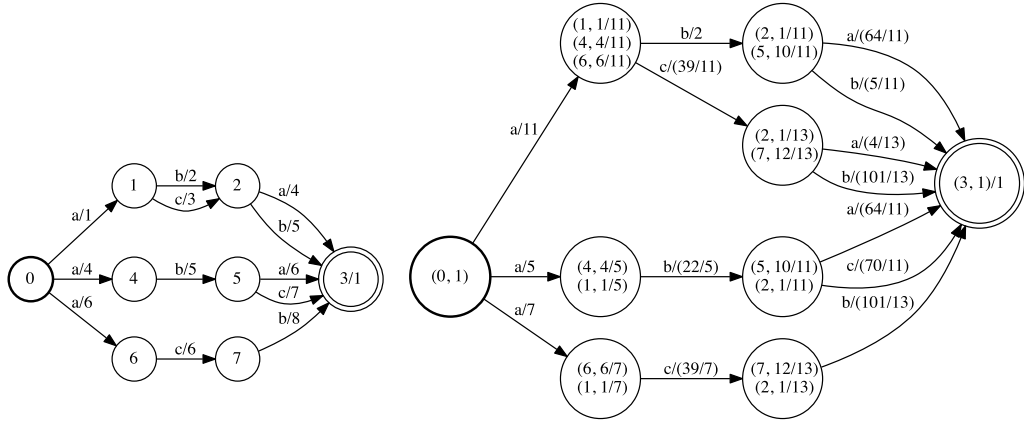
Fig. 2. Illustration of the R_f-pre-disambiguation construction in the semiring $(\mathbb{R}_+, +, \times, 0, 1)$. For each state (q, s) of the result, the subset s is explicitly shown. q is the state of the first pair in s shown. The weights are rational numbers, for example $\frac{1}{11} \approx .091$.

Note that the property $\mathrm{set}(s') = \delta_{\mathcal{A}}^{q'}(\mathrm{set}(s), a)$ always holds in the definition of the transition set $E_{\mathcal{B}}$ above. In particular, if p' is in $\delta_{\mathcal{A}}^{q'}(\mathrm{set}(s), a)$, then there is a path from $I_{\mathcal{A}}$ to some $p \in \mathrm{set}(s)$ labeled x and a transition from p to p' labeled with a and $p' R q'$ so p' is in $\mathrm{set}(s')$. Conversely, if p' is in $\mathrm{set}(s')$ then there exists p reachable by x with a transition labeled with a from p to p'. Since p' is in $\mathrm{set}(s')$, p' is in $\delta_{\mathcal{A}}^{q'}(I_{\mathcal{A}}, xa)$, thus $p' R q'$. Since there exists a transition labeled with a from q to q' and from p to p', this implies that $p R q$. Since $p R q$ and p is reachable via x, p is in $\delta_{\mathcal{A}}^{q}(I_{\mathcal{A}}, x)$.

When the set of states $Q_{\mathcal{B}}$ is finite, $\mathcal{B}$ is a WFA with a finite set of states and transitions and is defined as the result of the R-*pre-disambiguation of* $\mathcal{A}$. In general, R-pre-disambiguation is thus defined only for a subset of WFAs, which we will refer to as the set of R-*pre-disambiguable WFAs*. We will show later sufficient conditions for WFA $\mathcal{A}$ to be R-*pre-disambiguable* in the case of the tropical semiring. Fig. 2 illustrates the R-pre-disambiguation construction.

4.3. Properties of the resulting WFA

In this section, we assume that the input WFA $\mathcal{A}$ is R-pre-disambiguable that is that the number of states of the WFA $\mathcal{B}$ generated by R-pre-disambiguation is finite. Let us emphasize that the results presented in this section for R-pre-disambiguation hold for an arbitrary WFA over an admissible semiring for which the application of R-pre-disambiguation results in a finite number of states.

In general, the WFA $\mathcal{B}$ constructed by R-pre-disambiguation is not equivalent to $\mathcal{A}$, but the weight of each path in $\mathcal{B}$ starting at an initial state to some state (q, s) equals the $\oplus$-sum of the weights of every path in $\mathcal{A}$ with the same label starting at an initial state to a state in relation with q.

Proposition 1. *Let $\mathcal{B}$ be the WFA returned by the R-pre-disambiguation of the WFA $\mathcal{A}$. Then, the following equalities hold for any path $\pi \in P_{\mathcal{B}}(I_{\mathcal{B}}, x, (q, s))$ in $\mathcal{B}$, with $x \in \Sigma^*$ and $s = \{(p_1, w_1), \ldots, (p_t, w_t)\}$:*

$$w_{\mathcal{B}}^{\mathcal{I}}[\pi] = W_{\mathcal{A}}^{\mathcal{I}}(x, \mathrm{set}(s)) \quad \text{and} \quad \forall i \in [1, t], \ w_{\mathcal{B}}^{\mathcal{I}}[\pi] \otimes w_i = W_{\mathcal{A}}^{\mathcal{I}}(x, p_i).$$

The proof of is given in Appendix A.

Proposition 2. *Let $\mathcal{B}$ be the WFA returned by the R-pre-disambiguation of the WFA $\mathcal{A}$. Then, for any accepting path $\pi \in P_{\mathcal{B}}(I_{\mathcal{B}}, x, (q, s))$ in $\mathcal{B}$, with $x \in \Sigma^*$ and $(q, s) \in F_{\mathcal{B}}$, the following equality holds:*

$$w_{\mathcal{B}}^{\mathcal{I}}[\pi] \otimes \rho_{\mathcal{B}}((q, s)) = \mathcal{A}(x).$$

Proof. Let $s = \{(p_1, w_1), \ldots, (p_t, w_t)\}$. By the definition of $\rho_{\mathcal{B}}$, we can write

$$w_{\mathcal{B}}^{\mathcal{I}}[\pi] \otimes \rho_{\mathcal{B}}((q, s)) = w_{\mathcal{B}}^{\mathcal{I}}[\pi] \otimes \bigoplus_{\substack{p_i \in F_{\mathcal{A}} \\ i \in [1, t]}} (w_i \otimes \rho_{\mathcal{A}}(p_i)) = \bigoplus_{\substack{p_i \in F_{\mathcal{A}} \\ i \in [1, t]}} (w_{\mathcal{B}}^{\mathcal{I}}[\pi] \otimes w_i \otimes \rho_{\mathcal{A}}(p_i)).$$

Plugging in the expression of $(w^{\mathcal{I}}[\pi] \otimes w_i)$ given by Proposition 1 yields

$$w_{\mathcal{B}}^{\mathcal{I}}[\pi] \otimes \rho_{\mathcal{B}}((q, s)) = \bigoplus_{\substack{p_i \in F_{\mathcal{A}} \\ i \in [1, t]}} (W_{\mathcal{A}}^{\mathcal{I}}(x, p_i) \otimes \rho_{\mathcal{A}}(p_i)). \tag{2}$$
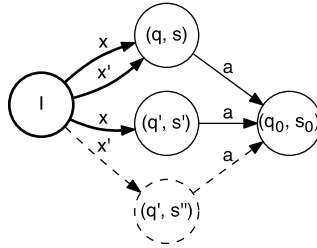
Fig. 3. Illustration of the proof of Lemma 1. The lemma proves the existence of the dashed transitions and the dashed state when $(q, s) \neq (q', s')$ and $x \neq x'$.

By the definition of R-pre-disambiguation, q is a final state. Any state $p \in \delta_{\mathcal{A}}(I, x) \cap F$ shares a common future with q since both p and q are final states, thus we must have $p \, R \, q$, which implies $p \in \mathrm{set}(s)$. Thus, the $\oplus$-sum in (2) is exactly over the set of states $\delta_{\mathcal{A}}(I, x) \cap F$, which proves that $w_{\mathcal{B}}^{\mathcal{I}}[\pi] \otimes \rho_{\mathcal{B}}((q, s)) = \mathcal{A}(x)$. $\square$

Proposition 3. *Let $\mathcal{B}$ be the WFA returned by the R-pre-disambiguation of the WFA $\mathcal{A}$. Then, any string $x \in \Sigma^*$ accepted by $\mathcal{A}$ is accepted by $\mathcal{B}$.*

Proof. Let $(q_0, a_1, w_1, q_1) \cdots (q_{n-1}, a_n, w_n, q_n)$ be an accepting path in $\mathcal{A}$ such that $a_1 \cdots a_n = x$. By construction, $((q_0, s_0), a_1, w'_1, (q_1, s_1)) \cdots ((q_{n-1}, s_{n-1}), a_n, w'_n, (q_n, s_n))$ is a path in $\mathcal{B}$ for some $w'_i \in \mathbb{S}$ and with $s_i = s(a_1 \cdots a_i, q_i)$ for all $i \in [1, n]$ and $s_0 = \epsilon$ and by the definition of finality in R-pre-disambiguation, (q_n, s_n) is final. Thus, x is accepted by $\mathcal{B}$. $\square$

5. Disambiguation algorithm

This section describes our disambiguation algorithm which applies to any WFA over an admissible semiring that is R-pre-disambiguable. Our algorithm consists of two steps: the R-pre-disambiguation algorithm presented and analyzed in the previous section, and a transition removal step and that requires a non-trivial proof of correctness.

We first present and analyze the transition removal step, next describe the full disambiguation algorithm, and illustrate it using several examples over different semirings.

5.1. Transition removal

Propositions 1, 2 and 3 show that the strings accepted by $\mathcal{B}$ are exactly those accepted by $\mathcal{A}$ and that the weight of any path in $\mathcal{B}$ accepting $x \in \Sigma^*$ is $\mathcal{A}(x)$. Thus, if for any x, we could eliminate from $\mathcal{B}$ all but one of the paths labeled with x, the resulting WFA would be unambiguous and equivalent to $\mathcal{A}$. Removing transitions to achieve this objective without changing the function represented by the WFA turns out not to be straightforward. The following two lemmas (Lemmas 1 and 2) and their proofs are the critical technical ingredients helping us define the transition removal and prove its correctness. This first lemma provides a useful tool for the proof of the second.

Lemma 1. *Let $\mathcal{B}$ be the WFA returned by the R-pre-disambiguation of the WFA $\mathcal{A}$. Let (q, s) and (q', s') be two distinct states of $\mathcal{B}$ both admitting a transition labeled with $a \in \Sigma$ to the same state (q_0, s_0) (or both final states), and such that $(q, s) \in \delta_{\mathcal{B}}(I_{\mathcal{B}}, x)$ and $(q', s') \in \delta_{\mathcal{B}}(I_{\mathcal{B}}, x)$ for some $x \in \Sigma^*$. Then, if $(q, s) \in \delta_{\mathcal{B}}(I_{\mathcal{B}}, x')$ for some $x' \neq x$, $x' \in \Sigma^*$, there exists a state $(q', s'') \in \delta_{\mathcal{B}}(I_{\mathcal{B}}, x')$ with $(q', s'') \neq (q, s)$ and such that (q', s'') admits a transition labeled with a to (q_0, s_0) (resp. is a final state).*

Proof. Fig. 3 illustrates the proof of the lemma. First, note that since $s = s(x, q)$ and $s' = s(x, q')$, $q = q'$ implies $(q, s) = (q', s')$. By contraposition, since $(q, s) \neq (q', s')$, we must have $q \neq q'$. Since both $q_0 \in \delta_{\mathcal{A}}(q, a)$ and $q_0 \in \delta_{\mathcal{A}}(q', a)$ in $\mathcal{A}$ (or both q and q' are final states), q and q' share a common future, which implies $q \, R \, q'$. Since (q', s') is reachable by x in $\mathcal{B}$ from $I_{\mathcal{B}}$, q' must be reachable by x from I in $\mathcal{A}$. This, combined with $q \, R \, q'$, implies that q' must be in $\mathrm{set}(s)$. Since $(q, s) \in \delta_{\mathcal{B}}(I_{\mathcal{B}}, x')$, all states in $\mathrm{set}(s)$ must be reachable by x' from I in $\mathcal{A}$, in particular q'. Thus, by definition of the R-pre-disambiguation construction, $\mathcal{B}$ admits a state $(q', s(q', x'))$, which is distinct from (q, s) since $q \neq q'$. If (q, s) admits a transition labeled with a to (q_0, s_0), then we have $s_0 = s(q_0, x'a)$. If (q', s') also admits a transition labeled with a to (q_0, s_0), then q' admits a transition labeled with a to q_0 and by the definition of the R-pre-disambiguation construction, $(q', s(q', x'))$ must admit a transition by a to $(q_0, s(q_0, x'a)) = (q_0, s_0)$. Finally, in the case where both (q, s) and (q', s') are final states, then q' is final in $\mathcal{A}$ and thus $(q', s(q', x'))$ is a final state in $\mathcal{B}$. $\square$

Let $\mathcal{B}$ be the WFA returned by the R-pre-disambiguation of the WFA $\mathcal{A}$. For any state (q_0, s_0) of $\mathcal{B}$ and label $a \in \Sigma$, let $\mathcal{L}(q_0, s_0, a) = ((q_1, s_1), \ldots, (q_n, s_n))$, $n \geq 1$, be the list of all distinct states of $\mathcal{B}$ admitting a transition labeled with $a \in \Sigma$ to (q_0, s_0), with $q_1 \leq \cdots \leq q_n$. This configuration is depicted in Fig. 4a. We define the *processing* of the list $\mathcal{L}(q_0, s_0, a)$ as
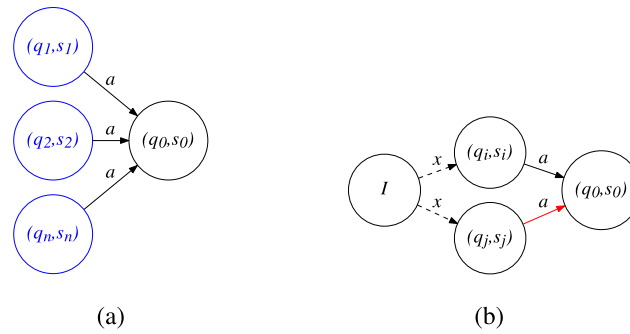
Fig. 4. *List Processing.* The three leftmost states in (a) depicted in blue represent the list $\mathcal{L}(q_0, s_0, a)$. The red-colored transition in (b) from state (q_j, s_j) will be removed since state (q_i, s_i) is also reachable by a path labeled with x, its index i is less than j and it has not already been removed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

follows: the states of the list are processed in order; for each state (q_j, s_j), $j \geq 2$, this consists of removing its a-transition to (q_0, s_0) if and only if there exists a co-reachable state (q_i, s_i) with $1 \leq i < j$ whose a-transition to (q_0, s_0) has not been removed.[2] This step is illustrated in Fig. 4b. Note that, by definition, the a-transition to (q_0, s_0) of the first state (q_1, s_1) is kept.

We define in a similar way the processing of the list $\mathcal{F} = ((q_1, s_1), \ldots, (q_n, s_n))$, $n \geq 1$, of all distinct final states of $\mathcal{B}$, with an arbitrary order $q_1 \leq \cdots \leq q_n$ as follows: the states of the list are processed in order; for each state (q_j, s_j), $j \geq 1$, this consists of making it non-final if and only if there exists a co-reachable state (q_i, s_i) with $i < j$ whose finality has been maintained. By definition, the finality of state (q_1, s_1) is maintained.

Lemma 2. *Let $\mathcal{B}$ be the WFA returned by the R-pre-disambiguation of the WFA $\mathcal{A}$. Let (q_0, s_0) be a state of $\mathcal{B}$ and $a \in \Sigma$, then, the WFA $\mathcal{C}$ resulting from processing the list $\mathcal{L}(q_0, s_0, a)$ accepts the same strings as $\mathcal{B}$. Similarly, the processing of the list of final states $\mathcal{F}$ of $\mathcal{B}$ does not affect the set of strings accepted by $\mathcal{B}$.*

Proof. Fix $a \in \Sigma$ and let $\mathcal{L}(q_0, s_0, a) = ((q_1, s_1), \ldots, (q_n, s_n))$, $n \geq 1$, be the list of all distinct states of $\mathcal{B}$ admitting a transition labeled with $a \in \Sigma$ to (q_0, s_0), with $q_1 \leq \cdots \leq q_n$. By definition, the a-transition of the first state (q_1, s_1) is kept, thus the set of strings accepted is unchanged after processing the first state. Assume now that the set of strings accepted is the same as that of $\mathcal{B}$ after processing all states $(q_1, s_1), \ldots, (q_i, s_i)$, $i \in [1, n-1]$. Assume that after processing (q_{i+1}, s_{i+1}) its a-transition to (q_0, s_0) is removed, otherwise the set of strings accepted is clearly unchanged and is thus the same as $\mathcal{B}$ by the induction hypothesis. The removal occurs because (q_{i+1}, s_{i+1}) and some state (q_j, s_j) are both in $\delta_{\mathcal{B}}(I_{\mathcal{B}}, x)$ for some $x \in \Sigma^*$, with $j < i+1$. The removal of the transition could potentially cause the elimination of a string accepted by the automaton because (q_{i+1}, s_{i+1}) may be reachable by some other string $x' \neq x$ that does not reach (q_j, s_j). Assume that (q_{i+1}, s_{i+1}) is reachable by such a string $x' \neq x$. We will show that at least one previously processed state is reachable by x' whose a-transition to (q_0, s_0) has not been removed. This will prove that the set of strings accepted is not affected by the processing of (q_{i+1}, s_{i+1}).

Assume that no such previously processed state exists. By Lemma 1, there exists a state (q_{k_1}, s_{k_1}) in $\mathcal{L}(q_0, s_0, a)$ reachable by x', distinct from (q_{i+1}, s_{i+1}) and with $q_{k_1} = q_j$. State (q_{k_1}, s_{k_1}) must have been processed before (q_{i+1}, s_{i+1}), otherwise, $j < i+1 < k_1$ and $q_{k_1} = q_j$ would imply $q_j = q_{i+1}$, which cannot be since, by construction, two distinct states of $\mathcal{B}$ of the form (q_j, s_j) and (q_j, s_j) cannot be co-reachable. Thus, since by assumption no previously processed state admitting a a-transition to (q_0, s_0) is reachable by x', the a-transition from (q_{k_1}, s_{k_1}) to (q_0, s_0) must have been removed. By the same assumption, the removal of the a-transition from (q_{k_1}, s_{k_1}) must be because $(q_{k_1}, s_{k_1}) \in \delta_{\mathcal{B}}(I_{\mathcal{B}}, x'')$ and $(q_l, s_l) \in \delta_{\mathcal{B}}(I_{\mathcal{B}}, x'')$ for some string $x'' \neq x'$ and some $l < k_1$, and because the a-transition of (q_l, s_l) to (q_0, s_0) has not been removed. By Lemma 1, this implies the existence of a state (q_{k_2}, s_{k_2}) in $\mathcal{L}(q_0, s_0, a)$ reachable by x', with $q_{k_2} = q_l$ and with (q_{k_2}, s_{k_2}) distinct from (q_{k_1}, s_{k_1}). As argued before, this implies that (q_{k_2}, s_{k_2}) has been processed before (q_{k_1}, s_{k_1}), therefore we have $k_2 < k_1$. Since (q_{k_2}, s_{k_2}) is reachable by x', by assumption, its a-transition to (q_0, s_0) must have been removed. Proceeding in this way, we can construct an infinite sequence of strictly decreasing indices $k_1 > k_2 > \ldots > k_m > \cdots$ of states (q_{k_m}, s_{k_m}) in $\mathcal{L}(q_0, s_0, a)$ reachable by x', which would contradict the finiteness of $\mathcal{L}(q_0, s_0, a)$. Thus, there exists a previously processed state in $\mathcal{L}(q_0, s_0, a)$ whose a-transition has not been removed and that is reachable by x', which concludes the proof of the first claim. The proof for processing $\mathcal{F}$ follows the same steps.[3] □

---

[2] This condition can in fact be relaxed: it suffices that there exists a co-reachable state (q_i, s_i) with $i < j$ since it can be shown that in that case, there exists necessarily such a state with a a-transition to (q_0, s_0).

[3] We can also introduce a *super-final state* f to which all final states of $\mathcal{B}$ are connected by a transition labeled with an auxiliary symbol $\phi \notin \Sigma$ with the semantics of finality. The proof is then syntactically the same as for regular symbols.
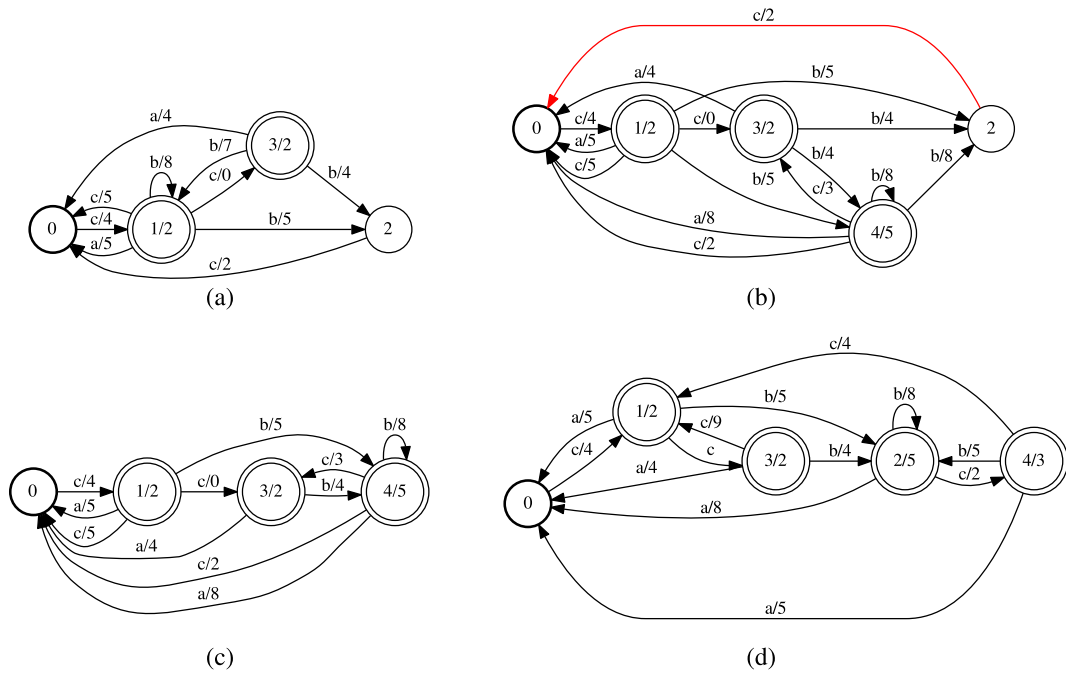
Fig. 5. Example illustrating the full disambiguation algorithm applied to a non-acyclic WFA. (a) WFA $\mathcal{A}$ over the tropical semiring. (b) WFA $\mathcal{B}$ obtained from $\mathcal{A}$ by application of pre-disambiguation. (c) WFA $\mathcal{C}$ result of our disambiguation algorithm applied to $\mathcal{A}$. $\mathcal{C}$ is obtained from $\mathcal{B}$ by removal of the red-colored transition from state 2 labeled with $c/2$ and trimming. (d) WFA obtained from $\mathcal{A}$ by application of determinization.

5.2. Disambiguation

Assume that $\mathcal{A}$ is R-pre-disambiguable. Then, this defines a disambiguation algorithm DISAMBIGUATION for $\mathcal{A}$ as follows:

(1) construct $\mathcal{B}$, the result of the R-pre-disambiguation of $\mathcal{A}$;
(2) for any state (q_0, s_0) of $\mathcal{B}$ and label $a \in \Sigma$, process $\mathcal{L}(q_0, s_0, a)$; process the list of final states $\mathcal{F}$.

Theorem 1. *Let $\mathcal{A}$ be a R-pre-disambiguable WFA over an admissible semiring. Then, algorithm DISAMBIGUATION run on input $\mathcal{A}$ generates an unambiguous WFA $\mathcal{B}$ equivalent to $\mathcal{A}$.*

Proof. Let $\mathcal{B}$ be the WFA returned by R-pre-disambiguation run with input $\mathcal{A}$. By Lemma 2, the set of strings accepted after processing the lists $\mathcal{L}(q_0, s_0, a)$ and $\mathcal{F}$ remains the same.[4] Furthermore, in view of the Propositions 1–3, the weight of the unique path labeled with an accepted string x in B $\otimes$-multiplied by its final weight is exactly $\mathcal{A}(x)$. Finally, by the definition of the processing operations, the resulting WFA is unambiguous, thus B is an unambiguous WFA equivalent to $\mathcal{A}$. □

Differing numberings of the states can lead to different orderings in each list and thus to different transition or finality removals, thereby resulting in different WFAs, with potentially different sizes after trimming. Nevertheless, all such resulting WFAs are equivalent.

Note that our disambiguation applies to any WFA $\mathcal{A}$ that is R-pre-disambiguable.

5.3. Disambiguation examples

Fig. 5 gives an example illustrating the pre-disambiguation and transition-removal stages of our disambiguation algorithm and also shows the result of determinization.

Fig. 6 gives another example illustrating the application of our disambiguation algorithm to a WFA on the semiring $(\mathbb{R}_+, +, \times, 0, 1)$.

Fig. 7 gives another example illustrating the application of our disambiguation algorithm to a tropical semiring WFA.

---

[4] The lemma is stated as processing one list, but from the proof it is clear it applies to multiple lists.
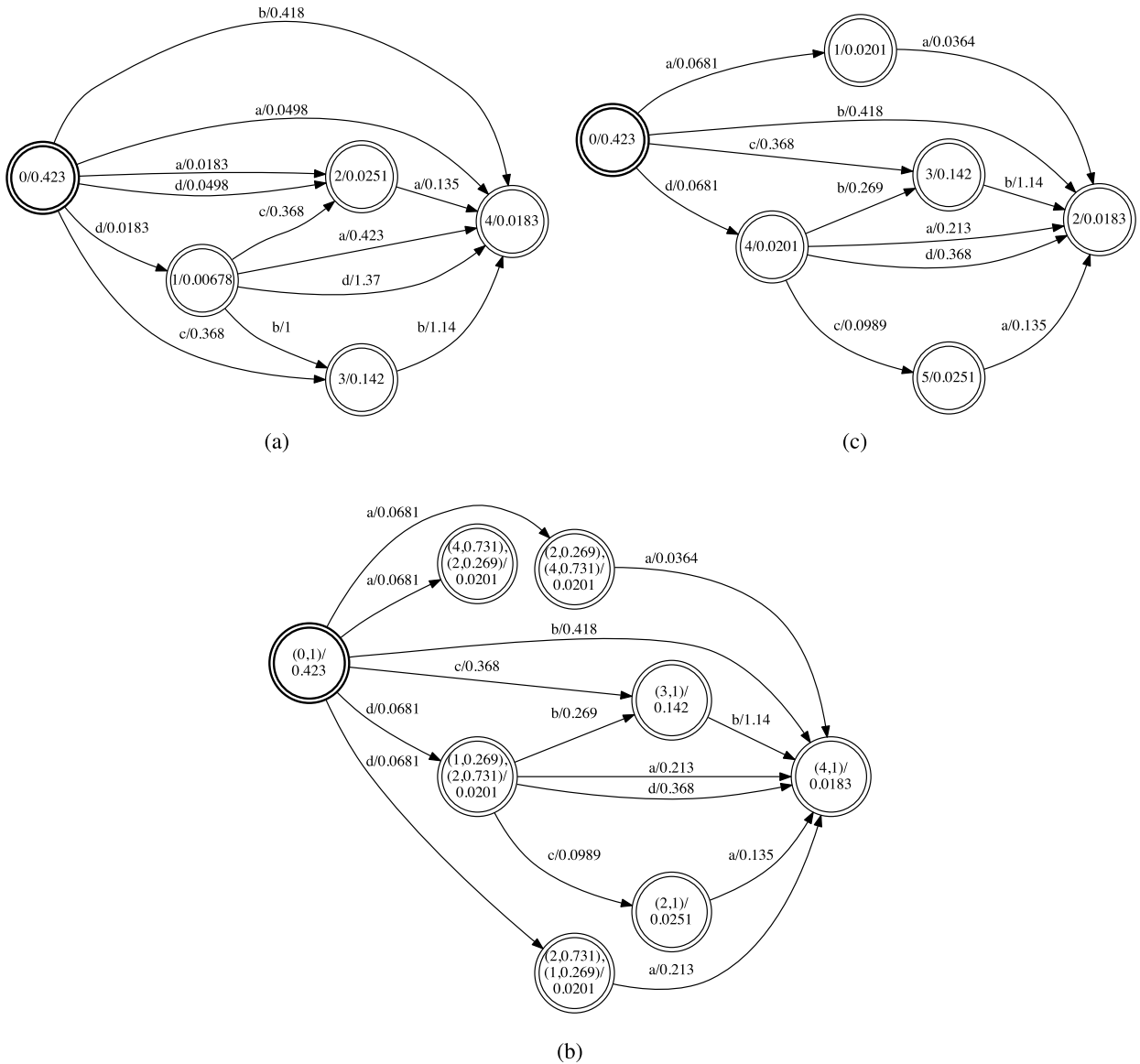
Fig. 6. Example illustrating the full disambiguation algorithm applied to a WFA defined on the semiring $(\mathbb{R}_+, +, \times, 0, 1)$. (a) input WFA $\mathcal{A}$ on this semiring. (b) WFA $\mathcal{B}$ obtained from $\mathcal{A}$ by application of R-pre-disambiguation. (c) WFA $\mathcal{C}$ result of our disambiguation algorithm applied to $\mathcal{A}$ obtained from $\mathcal{B}$ by transition removal.

5.4. Example of a non-disambiguable WFA

Some WFAs do not admit an equivalent unambiguous WFA. Fig. 8 shows an example of such a WFA defined over the tropical semiring, as shown by the following proposition. This example was also presented by [15] with a slightly different proof.

Proposition 4. *Let $\mathcal{A}_0$ be the WFA defined over the tropical semiring of Fig. 8. Then, there exists no deterministic WFA or unambiguous WFA over the tropical semiring equivalent to $\mathcal{A}_0$.*

Proof. It is not hard to see that $\mathcal{A}_0$ defines the following function,

$$\forall x \in \Sigma^*, \mathcal{A}_0(x) = \min\{|x|_a, |x|_b\}. \tag{3}$$

Assume that there exists a deterministic WFA $\mathcal{D}$ over the tropical semiring equivalent to $\mathcal{A}_0$. Consider the set of states in $\mathcal{D}$ reached when reading a^n, $n \in \mathbb{N}$ from the initial state. Since there are finitely many states, there exist $n_1, n_2 \in \mathbb{N}$, $n_1 < n_2$ such that a^{n_1} and a^{n_2} reach the same state $q \in Q$.
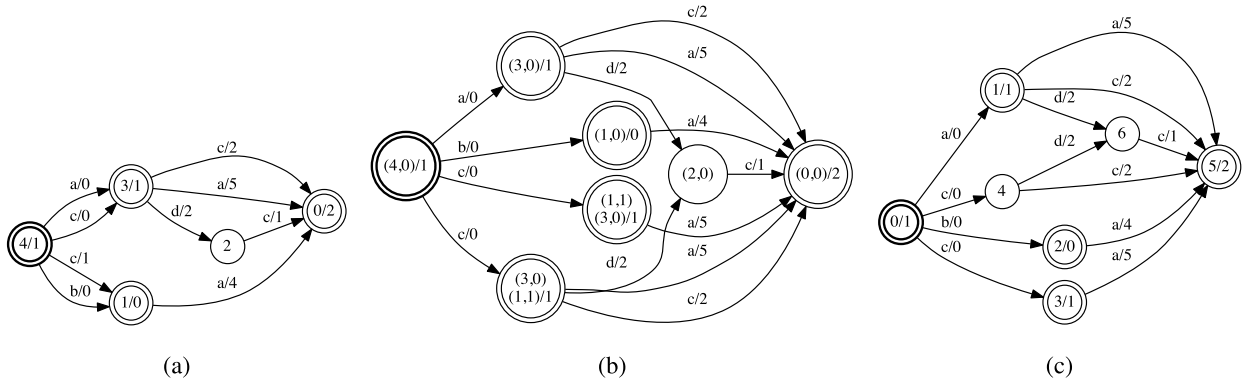
Fig. 7. Example illustrating the full disambiguation algorithm applied to a WFA defined over the tropical semiring. (a) WFA $\mathcal{A}$ over the tropical semiring. (b) WFA $\mathcal{B}$ obtained from $\mathcal{A}$ by application of R_f-pre-disambiguation. (c) WFA $\mathcal{C}$ result of our disambiguation algorithm applied to $\mathcal{A}$ obtained from $\mathcal{B}$ by transition removal.
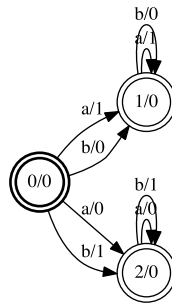

Fig. 8. Example of a non-disambiguable WFA $\mathcal{A}_0$ over the tropical semiring.

Let w_1 (w_2) denote the weight of the path from the initial state to q labeled with a^{n_1} (resp. a^{n_2}), including the initial weight, and ρ the final weight at q. Similarly let w' denote the weight of the path from q to F labeled with b^{n_1} and let ρ' be the final weight at the destination state of that path. By definition of $\mathcal{A}_0$, $\mathcal{A}_0(a^{n_1}b^{n_1}) = w_1 + w' + \rho' = n_1$ and $\mathcal{A}_0(a^{n_2}b^{n_1}) = w_2 + w' + \rho' = n_1$, thus $w_1 = w_2$. But, $\mathcal{A}_0(a^{n_1}) = w_1 + \rho = n_1$ and $\mathcal{A}_0(a^{n_2}) = w_2 + \rho = n_2$, which contradicts $w_1 = w_2$. Thus, there exists no deterministic WFA $\mathcal{D}$ equivalent to $\mathcal{A}_0$.

Assume now that there exists a trim unambiguous WFA $\mathcal{U}$ over the tropical semiring equivalent to $\mathcal{A}_0$. Then, the weight of a cycle in $\mathcal{U}$ labeled with a string x must be $\min\{|x|_a, |x|_b\}$. Thus, two cycles labeled with the same string have the same weight and $\mathcal{U}$ has the twins property. By [17] [Theorems 11 and 12], $\mathcal{U}$ is determinizable. This contradicts the non-existence of a deterministic WFA $\mathcal{D}$ equivalent to $\mathcal{A}_0$. Thus, $\mathcal{A}_0$ admits no equivalent unambiguous WFA $\mathcal{U}$. □

6. Sufficient conditions

6.1. Tropical semiring

The definition of siblings and that of twins property for WFAs were previously given by [17] (see also [3]). We will use a weaker (sufficient) condition for R-pre-disambiguability.

Definition 1. Two states p and q of a WFA $\mathcal{A}$ are said to be *siblings* if there exist two strings $x, y \in \Sigma^*$ such that both p and q can be reached from an initial state by paths labeled with x and there are cycles at both p and q labeled with y.

Two sibling states p and q are said to be *twins* if for any such x and y, $W_{\mathcal{A}}(p, y, p) = W_{\mathcal{A}}(q, y, q)$. $\mathcal{A}$ is said to have the *twins property* when any two siblings are twins. It is said to have the R-*weak twins property* when any two siblings that are in R relation are twins. When $\mathcal{A}$ admits the R_f-*weak twins property*, we will also say in short that it admits the *weak twins property*.

These definitions are illustrated in Fig. 9.

The results given in the remainder of this section are presented in the specific case of the tropical semiring. To show the following theorem we partly use a proof technique from [17] for showing that the twins property is a sufficient condition for weighted determinizability.

Theorem 2. Let $\mathcal{A}$ be a WFA over the tropical semiring that admits the R-weak twins property. Then, $\mathcal{A}$ is R-pre-disambiguable.
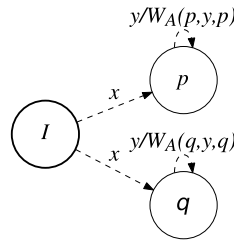
Fig. 9. States p and q are *siblings* since they both can be reached by paths labeled with x from the initial state and have cycles labeled by y. When $W_{\mathcal{A}}(p, y, p) = W_{\mathcal{A}}(q, y, q)$, they are *twins* and when $p \ R \ q$ is additionally required, they are *weak twins*.
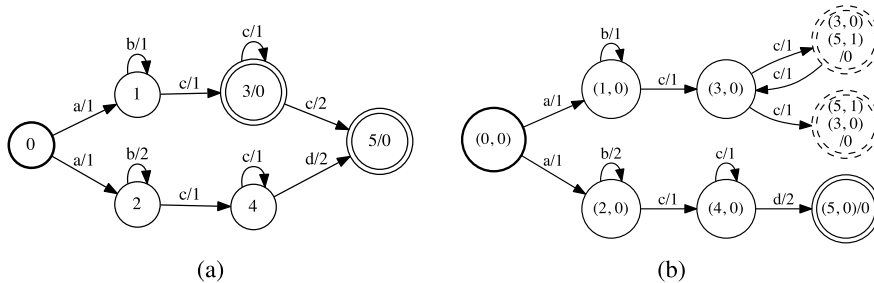

Fig. 10. (a) WFA $\mathcal{A}$ that cannot be determinized by the weighted determinization algorithm of [17]. (b) $\mathcal{A}$ has the weak twins property and can be disambiguated by DISAMBIGUATION as shown by the figure. One of the two states in dashed style is not made final by the algorithm. The head state for each of these states, is the state appearing in the first pair listed.

The proof is given in Appendix B. The theorem implies in particular that if $\mathcal{A}$ has the twins property then $\mathcal{A}$ is R-pre-disambiguable. In particular, any acyclic WFA is R-pre-disambiguable.

A WFA $\mathcal{A}$ is said to be *determinizable* when the weighted determinization algorithm of [17] terminates with input $\mathcal{A}$ (see also [3]). In that case, the output of the algorithm is a deterministic WFA equivalent to $\mathcal{A}$.

Theorem 3. *Let $\mathcal{A}$ be a determinizable WFA over the tropical semiring, then $\mathcal{A}$ is R-pre-disambiguable.*

The proof is given in Appendix C. By the results of [12], this also implies that any polynomially ambiguous WFA that has the *clones property* is R-pre-disambiguable and can be disambiguated using DISAMBIGUATION. There are however WFAs that are R-pre-disambiguable and thus can be disambiguated using DISAMBIGUATION but that cannot be determinized using the algorithm of [17]. Fig. 10 gives an example of such a WFA. To see that the WFA $\mathcal{A}$ of Fig. 10 cannot be determinized, consider instead B obtained from $\mathcal{A}$ by removing the transition from state 3 to 5. B is unambiguous and does not admit the twins property (cycles at states 1 and 2 have distinct weights), thus it is not determinizable by Theorem 12 of [17]. Weighted determinization creates infinitely many subsets of the form $\{(1, 0), (2, n)\}$, $n \in \mathbb{N}$, for paths from the initial state labeled with ab^n. Precisely the same subsets are created when applying determinization to $\mathcal{A}$.

On the tropical semiring, define $-\mathcal{A}$ as the WFA in which each non-infinite weight in $\mathcal{A}$ is replaced by its negation. The following result can be proven in a way that is similar to the proof of the analogous result for the twins property given by [3].[5]

Theorem 4. *Let $\mathcal{A}$ be a trim polynomially ambiguous WFA over the tropical semiring. Then, $\mathcal{A}$ has the weak twins property iff the weight of any cycle in $B = \text{TRIM}(A \cap (-A))$ is 0.*

This leads to an algorithm for testing the weak twins property for polynomially ambiguous WFAs in time $O(|Q_{\mathcal{A}}|^2 + |E_{\mathcal{A}}|^2)$, as in [3]. It was shown by [13] that the twins property is a decidable property that is PSPACE-complete for general WFAs over the tropical semiring. We conjecture that the same property holds for the weak twins property and in fact that a variant of the proof of [13] could be used, but we leave that to future work.

Fig. 11 illustrates a case where, in view of Theorems 3 and 4, the WFA is not pre-disambiguable, while it admits a straightforward equivalent unambiguous WFA.

---

[5] In [3], the authors use instead the terminology of *cycle-unambiguous* WFAs, which coincides with that of polynomially ambiguous WFAs.
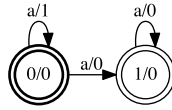
13

Fig. 11. Example of a WFA over the tropical semiring that is polynomially ambiguous and that does not admit the cycle weight property of Theorem 4. Since it does not admit weak twins property, it is not pre-disambiguable.

6.2. Finite semirings

The following is a straightforward result showing that our disambiguation can always be used when the admissible semiring is finite.

Theorem 5. *Let $\mathcal{A}$ be an arbitrary WFA over a finite admissible semiring. Then,* DISAMBIGUATION *can be applied to $\mathcal{A}$ and results in an equivalent unambiguous WFA.*

Proof. Since the semiring is finite, at most finitely many subsets can be created by R-pre-disambiguation. Thus, $\mathcal{A}$ is R-pre-disambiguable and DISAMBIGUATION can be used. □

This result can be extended to the case of locally finite semirings.

6.3. Other admissible semirings

We can define as with the twins property [3] an analogous notion of weak twins property for other admissible semirings than the tropical semiring. However, a result analogous to Theorem 2 does not hold in general.

Theorem 6. *Let $\mathcal{A}$ be an arbitrary acyclic WFA over an admissible semiring. Then,* DISAMBIGUATION *can be applied to $\mathcal{A}$ and results in an equivalent unambiguous WFA.*

Proof. Since the WFA is acyclic, at most finitely many subsets are created by R-pre-disambiguation. Thus, $\mathcal{A}$ is R-pre-disambiguable and DISAMBIGUATION can be used. □

7. Experiments

In order to experiment with weighted disambiguation, we implemented the algorithm (using the R_f relation) in the *OpenFst* C++ library [4]. For comparison, an implementation of weighted determinization is also available in that library [17].

For a first test corpus, we generated 500 speech *lattices* drawn from a randomized, anonymized utterance sampling of voice searches on the Google Android platform [21]. Each lattice is an acyclic WFA over spoken words that contains many weighted paths. Each path represents a hypothesis of what was uttered along with the automatic speech recognizer's (ASR) estimate of the probability of that path. Such lattices are useful for passing compact hypothesis sets to subsequent processing without commitment to, say, just one solution at the current stage.

The size of a lattice is determined by a probability threshold with respect to the most likely estimated path in the lattice; hypotheses within the threshold are retained in the lattice. Using $|\mathcal{A}| = |Q_{\mathcal{A}}| + |E_{\mathcal{A}}|$ to measure automata size, the mean size for these lattices was 2384 and the standard deviation was 3241.

The ASR lattices are typically non-deterministic and ambiguous due to both the models and the decoding strategies used. Determinization can be applied to reduce redundant computation in subsequent stages; disambiguation can be applied to determine the combined probability estimate of a string that may be distributed among several otherwise identically-labels paths.

Disambiguation has a mean expansion of 1.23 and a standard deviation of 0.59. Determinization has a mean expansion of 1.31 and a standard deviation of 1.35. For this data, disambiguation has a slightly lower mean expansion compared to determinization but a very substantially lower standard deviation.

As a second test corpus, we used 100 automata that are the compact representation of hypothesized Chinese-to-English translations from the DARPA Gale task [10]. These automata may contain cycles due to details of the particular translation system, which provides an interesting contrast to the acyclic speech case. Some fail to determinize within the allotted memory (1 GB) and about two-thirds of those also fail to disambiguate, possible when cycles are present.

Considering only those which are both determinizable and disambiguable, disambiguation has a mean expansion of 4.53 and a standard deviation of 6.0. Determinization has a mean expansion of 54.5 and a standard deviation of 90.5. For this data, disambiguation has a much smaller mean and standard deviation of expansion compared to determinization.

As a final example, Fig. 12 shows an acyclic unambiguous (unweighted) automaton whose size is in $O(n^2)$. No equivalent deterministic automaton can have fewer than 2^n states since such an automaton must have a distinct state for each of the prefixes of the strings $\{(a+b)^{k-1}b(a+b)^{n-k} : 1 \leq k \leq n\}$, which are prefixes of L. Thus, while our disambiguation algorithm
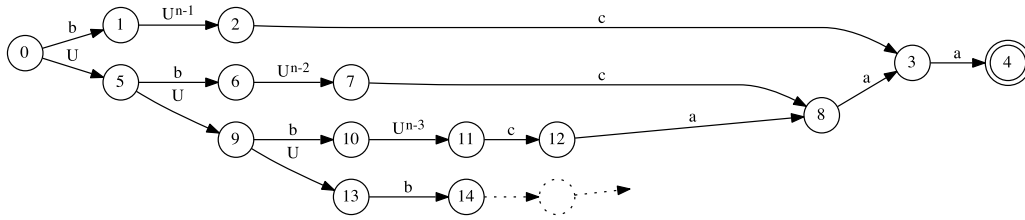
Fig. 12. Unambiguous automaton over the alphabet $\{a, b, c\}$ accepting the language $L = \{(a+b)^{k-1}b(a+b)^{n-k}ca^k : 1 \le k \le n\}$. For any $k \ge 0$, U^k serves as a shorthand for $(a+b)^k$.

leaves the automaton of Fig. 12 unchanged, determinization would result in this case in an automaton with more than 2^n states.

8. Conclusion

We presented an algorithm for the disambiguation of WFAs. The algorithm applies to a family of WFAs defined over the tropical semiring verifying a sufficient condition that we described, which includes all acyclic and, more generally, all determinizable WFAs. Our algorithm applies to WFAs defined over other admissible semirings that are R-pre-disambiguable. In particular, it applies to all acyclic WFAs defined over an arbitrary admissible semiring.

Our experiments showed the favorable properties of this algorithm in applications related to speech recognition and machine translation. The algorithm is likely to admit a large number of applications in areas such as natural language processing, speech processing, computational biology, and many other areas where WFAs are commonly used such as machine learning. The study of the theoretical properties we initiated raises a number of novel questions which include the following: the decidability of the weak twins property for arbitrary WFAs, the characterization of WFAs that admit an equivalent unambiguous WFA, the characterization of WFAs to which our algorithm can apply and perhaps an extension of our algorithm to a wider domain, and finally the proof and study of these questions for other semirings than the tropical semiring.

Acknowledgements

Appendix A. Proof of Proposition 1

Proof. The proof is by induction on the length of π. If π has length zero, it is a zero-length path from the state $(q, s(\epsilon, q)) \in I_\mathcal{B}$ to the same state and $w_\mathcal{B}^\mathcal{I}[\pi] = \lambda_\mathcal{B}((q, s))$. We have $W_\mathcal{A}^\mathcal{I}(\epsilon, \text{set}(s)) = \bigoplus_{\pi \in P_\mathcal{A}(I_\mathcal{A}, \epsilon, \text{set}(s))} w_\mathcal{A}^\mathcal{I}[\pi] = \bigoplus_{p \in \text{set}(s)} \lambda_\mathcal{A}(p) = \lambda_\mathcal{B}((q, s))$. Also, for all $i \in [1, t]$, $w_\mathcal{B}^\mathcal{I}[\pi] \otimes w_i = \lambda_\mathcal{B}((q, s)) \otimes \left[\lambda_\mathcal{B}((q, s))^{-1} \otimes \lambda_\mathcal{A}(p_i) \right] = \lambda_\mathcal{A}(p_i)$ and $W_\mathcal{A}^\mathcal{I}(\epsilon, p_i) = \lambda_\mathcal{A}(p_i)$, thus the equalities trivially hold.

Assume that the equalities hold for all paths of length at most $n \in \mathbb{N}$ starting in $I_\mathcal{B}$ and let π be a path of length $n+1$. We can therefore decompose as a path π in $P_\mathcal{B}(I_\mathcal{B}, x, (q', s'))$ for some $x \in \Sigma^*$, $q' \in Q_\mathcal{A}$, and $s' = \{(p'_1, w'_1), \ldots, (p'_{t'}, w'_{t'})\}$, followed by a transition $e = ((q', s'), a, w[e], (q, s))$ from (q', s') to (q, s). By definition of $w[e]$ in R-pre-disambiguation we can write

$$
w_\mathcal{B}^\mathcal{I}[\pi] = w_\mathcal{B}^\mathcal{I}[\pi'] \otimes w[e] = w_\mathcal{B}^\mathcal{I}[\pi'] \otimes \bigoplus_{j=1}^{t'} \left(w'_j \otimes W_\mathcal{A}(p'_j, a, \text{set}(s)) \right)
$$

$$
= \bigoplus_{j=1}^{t'} w_\mathcal{B}^\mathcal{I}[\pi'] \otimes w'_j \otimes W_\mathcal{A}(p'_j, a, \text{set}(s))
$$

$$
= \bigoplus_{j=1}^{t'} W_\mathcal{A}^\mathcal{I}(x, p'_j) \otimes W_\mathcal{A}(p'_j, a, \text{set}(s)), \tag{A.1}
$$

where we used the identities $w_\mathcal{B}^\mathcal{I}[\pi'] \otimes w'_j = W_\mathcal{A}^\mathcal{I}(x, p'_j)$, $j \in [1, t']$, which hold by the induction hypothesis.

We will show that any path ξ in $\mathcal{A}$ labeled with xa, starting in $I_\mathcal{A}$ and ending in $\text{set}(s)$ must go through $\text{set}(s')$, that is, ξ can be decomposed into a path labeled with x and reaching a state of $\text{set}(s')$ followed by a transition labeled with a from that state to a state of $\text{set}(s)$. (A.1) then implies that

$$
w_\mathcal{B}^\mathcal{I}[\pi] = W_\mathcal{A}^\mathcal{I}(xa, \text{set}(s)). \tag{A.2}
$$

Indeed, let $\xi = \xi' e'$ be a decomposition of ξ into a path ξ' labeled with x from $I_\mathcal{A}$ to some state $p' \in Q_\mathcal{A}$ followed by a transition e' labeled with a from p' to some state $p \in \mathrm{set}(s)$. By the definition of R-pre-disambiguation since p is in $\mathrm{set}(s)$, we have $p \mathrel{R} q$. By the compatibility of R with the inverse transition function, $p \in \delta_\mathcal{A}(p', a)$, and $q \in \delta_\mathcal{A}(q', a)$, this implies $p' \mathrel{R} q'$. Since we also have $p' \in \delta_\mathcal{A}(I_\mathcal{A}, x)$, this shows that p' is in $\mathrm{set}(s')$ and therefore that path ξ' ends in $\mathrm{set}(s')$.

In view of $w_\mathcal{B}^\mathcal{I}[\pi] = w_\mathcal{B}^\mathcal{I}[\pi'] \otimes w_\mathcal{B}[e]$ and using the definition of w_i in R-pre-disambiguation we can write, for any $i \in [1, t]$,

$$w_\mathcal{B}^\mathcal{I}[\pi] \otimes w_i = w_\mathcal{B}^\mathcal{I}[\pi'] \otimes w_\mathcal{B}[e] \otimes w_\mathcal{B}[e]^{-1} \otimes \Big(\bigoplus_{j=1}^{t'} w_j' \otimes W(p_j', a, p_i) \Big) \tag{A.3}$$

$$= w_\mathcal{B}^\mathcal{I}[\pi'] \otimes \Big(\bigoplus_{j=1}^{t'} w_j' \otimes W(p_j', a, p_i) \Big)$$

$$= \bigoplus_{j=1}^{t'} w_\mathcal{B}^\mathcal{I}[\pi'] \otimes w_j' \otimes W_\mathcal{A}(p_j', a, p_i) = \bigoplus_{j=1}^{t'} W_\mathcal{A}^\mathcal{I}(x, p_j') \otimes W(p_j', a, p_i),$$

using the identities $w_\mathcal{B}^\mathcal{I}[\pi'] \otimes w_j' = W_\mathcal{A}^\mathcal{I}(x, p_j')$, $j \in [1, t']$, which hold by the induction hypothesis.

By the same argument as the one already presented, a path ξ starting in $I_\mathcal{A}$ labeled with xa and ending in p_i must reach a state of $\mathrm{set}(s')$ after reading x. In view of that, (A.3) implies that

$$w_\mathcal{B}^\mathcal{I}[\pi] \otimes w_i = W_\mathcal{A}^\mathcal{I}(xa, p_i), \tag{A.4}$$

which concludes the proof. □

Appendix B. Proof of Theorem 2

Proof. Assume that $\mathcal{A}$ admits the R-*weak twins property* and that the R-pre-disambiguation construction creates infinitely many distinct states (q, s). Since the states defining the weighted subsets s are of a finite number, there must be infinitely many states (q_n, s_n), $n \in \mathbb{N}$, with the same $\mathrm{set}(s_n)$. Among these states, we must have $q_n = q$ for at least one state q for infinitely many $n \in \mathbb{N}$, since the number of distinct states q_n is finite. Thus, the assumption made implies that the R-pre-disambiguation construction creates an infinite sequence (q, s_n), $n \in \mathbb{N}$, with the same $\mathrm{set}(s_n) = \{p_1, \ldots, p_k\}$, $k < +\infty$, and say $p_1 = q$. Thus, we can write $s_n = \{(p_1, w_n(p_1)), \ldots, (p_k, w_n(p_k))\}$.

By Proposition 1, for any $n \in \mathbb{N}$, there exists a string $x_n \in \Sigma^*$, with

$$\forall p \in \{p_1, \ldots, p_k\}, \quad w_n(p) = W_\mathcal{A}^\mathcal{I}(x_n, p) - W_\mathcal{A}^\mathcal{I}(x_n, \{p_1, \ldots, p_k\}). \tag{B.1}$$

There exists at least one $p \in \{p_1, \ldots, p_k\}$ such that $W_\mathcal{A}^\mathcal{I}(x_n, \{p_1, \ldots, p_k\}) = W_\mathcal{A}^\mathcal{I}(x_n, p)$ for infinitely many indices $J \subseteq \mathbb{N}$, since k is finite. By (B.1), $w_n(p) = 0$ for all $n \in J$. $\{w_n(q) - w_n(p_i) : n \in J\}$ cannot be finite for all $i \in [1, k]$, otherwise in particular $\{w_n(q) - w_n(p) : n \in J\} = \{w_n(q) : n \in J\}$ would be finite, which in turn, by the finiteness of $\{w_n(q) - w_n(p_i) : n \in J\}$ for all i, would imply the finiteness $\{w_n(p_i) : n \in J\}$ for all i, contradicting the infiniteness of $\{s_n : n \in J\}$. Thus, there must exist at least one state $r \in \{p_1, \ldots, p_k\}$ such that $\{w_n(q) - w_n(r) : n \in J\}$ is infinite.

We will show that $\{w_n(q) - w_n(r) : n \in J\}$ is included in the finite set

$$A = \{w_\mathcal{A}^\mathcal{I}[\pi_1] - w_\mathcal{A}^\mathcal{I}[\pi_0] : \pi_1 \in P_\mathcal{A}(I_\mathcal{A}, x, q), \pi_0 \in P_\mathcal{A}(I_\mathcal{A}, x, r), |x| \leq |Q_\mathcal{A}|^2 - 1\}, \tag{B.2}$$

thereby contradicting the original assumption about R-pre-disambiguation creating infinitely many states.

Refer to a shortest path with an origin at an initial state and that includes the initial state's weight as an $\mathcal{I}$-*shortest-path*. Consider $x = x_n$ for some $n \in \mathbb{N}$. Let π_1 be an $\mathcal{I}$-shortest-path among $P_\mathcal{A}(I_\mathcal{A}, x, q)$ and π_0 an $\mathcal{I}$-shortest-path among $P_\mathcal{A}(I_\mathcal{A}, x, r)$. Thus, by (B.1), we can write

$$w_n = (w_\mathcal{A}^\mathcal{I}[\pi_1] - W_\mathcal{A}^\mathcal{I}(x, p)) - (w_\mathcal{A}^\mathcal{I}[\pi_0] - W_\mathcal{A}^\mathcal{I}(x, p)) = w_\mathcal{A}^\mathcal{I}[\pi_1] - w_\mathcal{A}^\mathcal{I}[\pi_0]. \tag{B.3}$$

Since both q and r are reachable from I by a path labeled with x, there is a path in $A \cap A$ from a pair of initial states to (q, r). Assume that $|x| > |Q_\mathcal{A}|^2 - 1$, then this path must go through at least one non-empty cycle at some state (q_1, r_1). Thus, by the definition of intersection, paths π_0 and π_1 can be decomposed as

$$\pi_1 = \pi_1^1 \pi_1^2 \pi_1^3 \quad \text{with } \pi_1^1 \in P_\mathcal{A}(I_\mathcal{A}, x^1, q_1), \pi_1^2 \in P_\mathcal{A}(q_1, x^2, q_1), \pi_1^2 \in P_\mathcal{A}(q_1, x^3, q)$$

$$\pi_0 = \pi_0^1 \pi_0^2 \pi_0^3 \quad \text{with } \pi_0^1 \in P_\mathcal{A}(I, x^1, r_1), \pi_0^2 \in P_\mathcal{A}(r_1, x^2, r_1), \pi_0^2 \in P_\mathcal{A}(r_1, x^3, r).$$

Since π_0 and π_1 are shortest paths, the cycles at q_1 and r_1 are also shortest paths. Now, by definition of the states created by R-pre-disambiguation all states in $\{p_1, \ldots, p_k\}$, in particular r, are in R-relation with q. By compatibility with the inverse

transition function, this implies that r_1 and q_1 are also in R-relation. Thus, by the R-weak twins property, the weight of the cycle at q_1 and that of the cycle at state r_1 in the decompositions above must be equal. Therefore, we can write

$$w_n = w_{\mathcal{A}}^{\mathcal{I}}[\pi_1'] - w_{\mathcal{A}}^{\mathcal{I}}[\pi_0'] \tag{B.4}$$

with $\pi_1' = \pi_1^1 \pi_1^3$ and $\pi_0' = \pi_0^1 \pi_0^3$. We have $|\pi_1'| < |\pi_1|$ and $|\pi_0'| < |\pi_0|$. Thus, by induction on $|x|$, we can find two paths $\pi_1'' \in P_{\mathcal{A}}(I_{\mathcal{A}}, x'', q)$ and $\pi_0'' \in P_{\mathcal{A}}(I_{\mathcal{A}}, x'', r)$ with $w_n = w_{\mathcal{A}}^{\mathcal{I}}[\pi_1''] - w_{\mathcal{A}}^{\mathcal{I}}[\pi_0'']$ and $|x''| \le |Q_{\mathcal{A}}|^2 - 1$. Proceeding in the same way for all x_n, this shows that $\{w_n : n \in J\}$ is included in the finite set $\mathcal{A}$, which contradicts the fact the number of states created by the R-pre-disambiguation construction is infinite. □

Appendix C. Proof of Theorem 3

Proof. Let $\mathcal{A}$ be a determinizable WFA. Assume that the application of R-pre-disambiguation to $\mathcal{A}$ generates an infinite set of distinct states. Then, as in the proof of Theorem 2, this implies the existence of two states q and r reachable from the initial states by strings $(x_n)_{n \in \mathbb{N}}$, and such that the set $\{w_n(q) - w_n(r) : n \in \mathbb{N}\} = \{W_{\mathcal{A}}^{\mathcal{I}}(x_n, q) - W_{\mathcal{A}}^{\mathcal{I}}(x_n, r) : n \in \mathbb{N}\}$ is infinite.

For any $n \in \mathbb{N}$, consider the weighted subset S_n constructed by weighted determinization which is the set of pairs (p, v), where p is a state of $\mathcal{A}$ reachable by x_n from the initial state and v its *residual weight* defined by $v = W_{\mathcal{A}}^{\mathcal{I}}(x_n, p) - \min_{p' \in \delta_S crA(I, x_n)} W_{\mathcal{A}}^{\mathcal{I}}(x_n, p')$. Since $\mathcal{A}$ is determinizable, there can only be finitely many distinct S_n, $n \in \mathbb{N}$. S_n includes the pairs (q, v_n) and (r, v_n') with $v_n = W_{\mathcal{A}}^{\mathcal{I}}(x_n, q) - v_0$ and $v_n' = W_{\mathcal{A}}^{\mathcal{I}}(x_n, q) - v_0$, where v_0 is the weight of an $\mathcal{I}$-shortest-path labeled with x_n and starting at the initial states. Since the number of distinct weighted subsets S_n is finite, so must be the number of distinct pairs $((q, v_n), (r, v_n'))$ they each include. This implies that there are only finitely many distinct differences of weight in $\{v_n' - v_n : n \in I\}$. But, since $v_n' - v_n = W_{\mathcal{A}}^{\mathcal{I}}(x_n, q) - W_{\mathcal{A}}^{\mathcal{I}}(x_n, r)$, this contradicts the infiniteness of $\{W_{\mathcal{A}}^{\mathcal{I}}(x_n, q) - W_{\mathcal{A}}^{\mathcal{I}}(x_n, r) : n \in \mathbb{N}\}$. Thus, R-pre-disambiguation cannot generate an infinite number of states and $\mathcal{A}$ is R-pre-disambiguable. □

References

[1] J. Albert, J. Kari, Digital image compression, in: Handbook of Weighted Automata, Springer, 2009, pp. 453–479.
[2] C. Allauzen, E. Benson, C. Chelba, M. Riley, J. Schalkwyk, Voice query refinement, in: Interspeech, 2012, pp. 2462–2465.
[3] C. Allauzen, M. Mohri, Efficient algorithms for testing the twins property, J. Autom. Lang. Comb. 8 (2) (2003) 117–144.
[4] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, M. Mohri, OpenFst library, http://www.openfst.org, 2007.
[5] T.M. Breuel, The OCRopus open source OCR system, in: Electronic Imaging 2008, International Society for Optics and Photonics, 2008, Article 68150F.
[6] C. Cortes, V. Kuznetsov, M. Mohri, M.K. Warmuth, On-line learning algorithms for path experts with non-additive losses, in: Proceedings of COLT, 2015, pp. 424–447.
[7] R. Durbin, S.R. Eddy, A. Krogh, G.J. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Camb. Univ. Press, 1998.
[8] S. Eilenberg, Automata, Languages and Machines, Academic Press, 1974.
[9] D. Eppstein, Finding the k shortest paths, SIAM J. Comput. 28 (2) (1998) 652–673.
[10] G. Iglesias, C. Allauzen, W. Byrne, A. de Gispert, M. Riley, Hierarchical phrase-based translation representations, in: Proceedings of EMNLP, 2011, pp. 1373–1383.
[11] R.M. Kaplan, M. Kay, Regular models of phonological rule systems, Comput. Linguist. 20 (3) (1994).
[12] D. Kirsten, A Burnside approach to the termination of Mohri's algorithm for polynomially ambiguous min-plus-automata, RAIRO Theor. Inform. Appl. 42 (3) (2008) 553–581.
[13] D. Kirsten, Decidability, undecidability, and PSPACE-completeness of the twins property in the tropical semiring, Theoret. Comput. Sci. 420 (2012) 56–63.
[14] D. Kirsten, S. Lombardy, Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata, in: STACS, 2009, pp. 589–600.
[15] I. Klimann, S. Lombardy, J. Mairesse, C. Prieur, Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton, Theoret. Comput. Sci. 327 (3) (2004).
[16] W. Kuich, A. Salomaa, Semirings, Automata, Languages, EATCS Monographs on Theoretical Computer Science, vol. 5, Springer, Berlin, Germany, 1986.
[17] M. Mohri, Finite-state transducers in language and speech processing, Comput. Linguist. 23 (2) (1997) 269–311.
[18] M. Mohri, On the disambiguation of finite automata and functional transducers, Internat. J. Found. Comput. Sci. 24 (6) (2013) 847–862.
[19] M. Mohri, F.C.N. Pereira, M. Riley, Speech recognition with weighted finite-state transducers, in: L. Rabiner, F. Juang (Eds.), Handbook on Speech Processing and Speech Communication, Part E: Speech Recognition, Springer-Verlag, Heidelberg, Germany, 2008, pp. 559–582.
[20] M. Mohri, M. Riley, An efficient algorithm for the n-best-strings problem, in: Proceedings of ICSLP, INTERSPEECH, 2002, pp. 1313–1316.
[21] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, B. Strope, "Your word is my command": Google search by voice: a case study, in: Advances in Speech Recognition., Springer, 2010, pp. 61–90.
[22] E.M. Schmidt, Succinctness of description of context-free, regular and unambiguous languages, Ph.D. thesis, Dept. of Comp. Sci., University of Aarhus, 1978.