



**HAL**  
open science

# On Determinism and Unambiguity of Weighted Two-way Automata

Vincent Carnino, Sylvain Lombardy

► **To cite this version:**

Vincent Carnino, Sylvain Lombardy. On Determinism and Unambiguity of Weighted Two-way Automata. 14th International Conference on Automata and Formal Languages, May 2014, Szeged, Hungary. pp.188-200, 10.4204/EPTCS.151.13 . hal-01008785

**HAL Id: hal-01008785**

**<https://hal.science/hal-01008785>**

Submitted on 19 Nov 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On Determinism and Unambiguity of Weighted Two-way Automata

Vincent Carnino

LIGM - Laboratoire d'informatique Gaspard-Monge  
Université Paris-Est Marne-la-Vallée, France

Vincent.Carnino@univ-mlv.fr

Sylvain Lombardy

LaBRI - Laboratoire Bordelais de Recherche en Informatique  
Institut Polytechnique de Bordeaux, France

Sylvain.Lombardy@labri.fr

In this paper, we first study the conversion of weighted two-way automata to one-way automata. We show that this conversion preserves the unambiguity but does not preserve the determinism. Yet, we prove that the conversion of an unambiguous weighted one-way automaton into a two-way automaton leads to a deterministic two-way automaton. As a consequence, we prove that unambiguous weighted two-way automata are equivalent to deterministic weighted two-way automata in commutative semirings.

## 1 Introduction

A classical question in automata theory concerns the expressive power of a device and especially the difference between one-way devices and two-way devices. It is well known that two-way automata may be reduced to one-way automata and therefore recognize the same language family [14, 12].

In this paper, we deal with the weighted versions of these two devices. We describe the conversion of a two-way automaton over a commutative semiring into a one-way automaton. Such an algorithm has already been stated in [1]; our construction is close, but we are mainly interested here in proving that this conversion preserves the unambiguity of automata; it does not preserve the determinism.

We then present a construction for the conversion of any unambiguous one-way automaton into a deterministic two-way automaton; this part does not require that the semiring is commutative.

A consequence of these two procedures is that, on commutative semirings, opposite to the case of one-way automata, unambiguous two-way automata are not more powerful than deterministic ones.

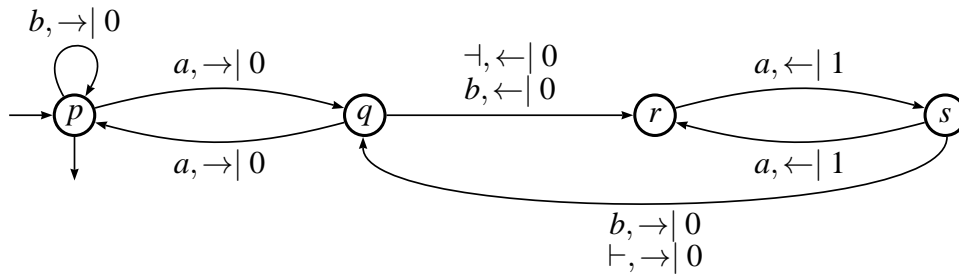
## 2 Weighted Two-way Automata

### 2.1 Automata and runs

A semiring  $\mathbb{K}$  is a set endowed with two binary associative operations,  $\oplus$  and  $\otimes$ , such that  $\oplus$  is commutative and  $\otimes$  distributes over  $\oplus$ . The set  $\mathbb{K}$  contains two particular elements,  $0_{\mathbb{K}}$  and  $1_{\mathbb{K}}$  that are respectively neutral for  $\oplus$  and  $\otimes$ ; moreover,  $0_{\mathbb{K}}$  is an annihilator for  $\otimes$ .

For every alphabet  $A$ , we assume that there exist two fresh symbols  $\vdash$  and  $\dashv$  that are marks at the beginning and the end of the tapes of automata. We denote  $A_{\vdash \dashv}$  the alphabet  $A \cup \{\vdash, \dashv\}$ . For every word  $w$  in  $A$ ,  $w_{\vdash \dashv}$  is the word in  $A_{\vdash \dashv}$  equal to  $\vdash w \dashv$ .

One-way and two-way  $\mathbb{K}$ -automata share a part of their definition. A  $\mathbb{K}$ -automaton is a tuple  $\mathcal{A} = (Q, A, E, I, T)$  where  $Q$  is a finite set of states,  $A$  is a finite alphabet, and  $I$  and  $T$  are partial functions from  $Q$  to  $\mathbb{K}$ . The support of  $I$ ,  $\underline{I}$ , is the set of initial states of  $\mathcal{A}$ , and the support of  $T$ ,  $\underline{T}$ , is the set of final states of  $\mathcal{A}$ .

Figure 1: The two-way  $\mathcal{N}$ -automaton  $\mathcal{A}_1$ .

The definition of transitions differ. In a two-way  $\mathbb{K}$ -automaton,  $E$  is a partial function from  $Q \times (A_{\vdash} \times \{-1, +1\}) \times Q$  into  $\mathbb{K}$  and the support of  $E$ ,  $\underline{E}$ , is the set of transitions of  $\mathcal{A}$ . Moreover, the intersection of  $\underline{E}$  and  $Q \times (\{\vdash\} \times \{-1\} \cup \{\dashv\} \times \{1\}) \times Q$  must be empty.

Let  $t$  be a transition in  $\underline{E}$ ; if  $t = (p, a, d, q)$ , we denote  $\sigma(t) = p$ ,  $\tau(t) = q$ ,  $\lambda(t) = a$ ,  $\delta(t) = d$ . On figures, the value of  $\delta$  is represented by a left (-1) or right (+1) arrow. For instance, if  $t = (p, a, -1, q)$  and  $E_t = k$ , we draw  $p \xrightarrow{a, \leftarrow | k} q$ .

In a one-way  $\mathbb{K}$ -automaton,  $E$  is a partial function from  $Q \times A \times Q$  into  $\mathbb{K}$ , and the support of  $E$ ,  $\underline{E}$ , is the set of transitions of  $\mathcal{A}$ .

Let  $t$  be a transition in  $\underline{E}$ ; if  $t = (p, a, q)$ , we denote  $\sigma(t) = p$ ,  $\tau(t) = q$ ,  $\lambda(t) = a$ .

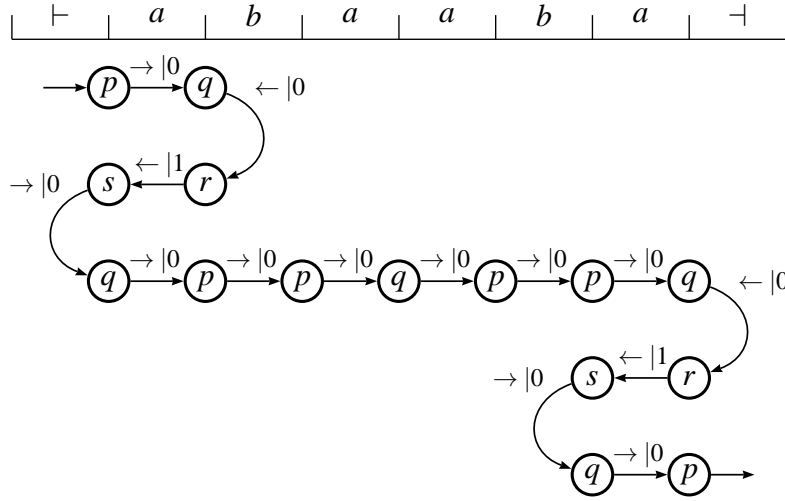
**Example 1.** Let  $\mathcal{A}_1$  be the two-way  $\mathcal{N}$ -automaton of Figure 1, where  $\mathcal{N} = (\mathbb{N} \cup \{\infty\}, \min, +)$  is the tropical semiring; since the multiplication law in this semiring is the usual sum, the weight of a path in this automaton is the sum of the weights of its transitions. This automaton is deterministic (cf. Definition 9) and thus there is only one computation for each accepted word. The behaviour of this automaton is quite easy. For each block of  $a'$  it checks through a left-right reading, whether the length of the block is odd; if it is, a right-left reading computes the length of the block; otherwise the automaton goes to the next block of  $a'$ .

**Definition 1.** Let  $w = w_1 \dots w_n$  be a word of  $A^*$ , we set  $w_0 = \vdash$  and  $w_{n+1} = \dashv$ . A configuration of  $\mathcal{A}$  on  $w$  is a pair  $(p, i)$  where  $i$  is in  $[0; n+1]$  and  $p$  is a state of  $\mathcal{A}$ . A computation (or run)  $\rho$  of  $\mathcal{A}$  on  $w$  is a finite sequence of configurations  $((p_0, i_0), \dots, (p_k, i_k))$  such that :

- $i_0 = 1$ ,  $i_k = n + 1$ ,  $p_0$  is in  $\underline{I}$  and  $p_k$  is in  $\underline{T}$ ;
- for every  $j$  in  $[0; k - 1]$ , there exists  $t_j$ , such that  $\sigma(t_j) = p_j$ ,  $\tau(t_j) = p_{j+1}$ ,  $\lambda(t_j) = a_{i_j}$ , and  $i_{j+1} = i_j + \delta(t_j)$ .

The weight of such a computation, denoted by  $|\rho|$ , is  $I(p_0) \otimes \bigotimes_{j=0}^{k-1} E(t_j) \otimes T(p_k)$ . The weight of  $w$  in  $\mathcal{A}$ , denoted by  $\langle |\mathcal{A}|, w \rangle$ , is the addition of the weights of all the runs with label  $w$  in  $\mathcal{A}$ . Notice that there may be an infinite number of computations with the same label  $w$ . The definition of the behaviour of  $\mathcal{A}$  in this case requires to study the definition of infinite sums. This can be done, like for one-way  $\mathbb{K}$ -automata with  $\varepsilon$ -transitions, for instance with complete semirings or topological semirings [11]. This is not the purpose of this paper, since we mainly deal with two-way automata where the number of computations is finite for every word.

**Example 2.** A run of the  $\mathcal{N}$ -automaton  $\mathcal{A}_1$  over the word  $abaaba$  is represented on Figure 2. The weight of this run is equal to 2.

Figure 2: A run of  $\mathcal{A}_1$  over the word  $abaaba$ .

**Definition 2.** Let  $\rho = ((p_0, i_0), \dots, (p_k, i_k))$  be a run over  $w$ . If there exists  $m, n$  in  $[1, k]$ , with  $m < n$  such that  $(p_m, i_m) = (p_n, i_n)$ , then we say that  $((p_m, i_m), \dots, (p_n, i_n))$  is an unmoving circuit of  $\rho$ . If  $\rho$  does not contain any unmoving circuit, it is reduced.

**Lemma 1.** If a two-way  $\mathbb{K}$ -automaton admits a run  $\rho$  which is not reduced, it admits a reduced run with the same label.

*Proof.* We consider a shortest non reduced run  $\rho = ((p_0, i_0), \dots, (p_m, i_m), \dots, (p_n, i_n), \dots, (t_k, i_k))$ , with  $(p_m, i_m) = (p_n, i_n)$ .

Then  $((p_0, i_0), \dots, (p_{m-1}, i_{m-1}), (p_n, i_n), \dots, (p_k, i_k))$  is a run; by minimality of  $\rho$ , this run is reduced.  $\square$

**Definition 3.** A one-way or two-way automaton  $\mathcal{A}$  is unambiguous if every word labels at most one computation.

Unambiguous automata have obviously only reduced computations.

## 2.2 Coverings

We extend here the notion of covering (cf. [13]) to two-way automata.

**Definition 4.** Let  $\mathcal{A} = (Q, A, E, I, T)$  and  $\mathcal{B} = (R, A, F, J, U)$  be two weighted two-way automata. A mapping  $\varphi$  from  $Q$  into  $R$  is a morphism if,

i)  $\forall p \in \underline{L}, J(\varphi(p)) = I(p)$ ;

ii)  $\forall p \in \underline{T}, U(\varphi(p)) = T(p)$ ;

iii)  $\forall t = (p, a, \delta, q) \in \underline{E}, \tilde{\varphi}(t) = (\varphi(p), a, \delta, \varphi(q)) \in \underline{F}$  and  $F(\tilde{\varphi}(t)) = E(t)$ .

The morphism is surjective if  $\varphi(Q) = R$ ,  $\varphi(\underline{L}) = \underline{J}$ ,  $\varphi(\underline{T}) = \underline{U}$ , and  $\tilde{\varphi}(\underline{E}) = \underline{F}$ .

**Definition 5.** Let  $\mathcal{A} = (Q, A, E, I, T)$  and  $\mathcal{B} = (R, A, F, J, U)$  be two weighted two-way automata.  $\mathcal{A}$  is a covering of  $\mathcal{B}$  if there exists a surjective morphism  $\varphi$  from  $\mathcal{A}$  onto  $\mathcal{B}$  such that

i)  $\forall r \in \underline{U}, \varphi^{-1}(r) \subseteq \underline{T}$     ii)  $\forall r \in \underline{J}, \exists! p \in \varphi^{-1}(r) \cap \underline{L}$

iii)  $\forall t \in \underline{F}, \forall p \in \varphi^{-1}(\sigma(t)), \exists! t' \in \tilde{\varphi}^{-1}(t), \sigma(t') = p$ .

$\mathcal{A}$  is an in-covering of  $\mathcal{B}$  is there exists a surjective morphism  $\varphi$  from  $\mathcal{A}$  onto  $\mathcal{B}$  such that

$$\begin{aligned} & i) \forall r \in \underline{J}, \varphi^{-1}(r) \subseteq \underline{I} \quad ii) \forall r \in \underline{U}, \exists! p \in \varphi^{-1}(r) \cap \underline{T} \\ & iii) \forall t \in \underline{E}, \forall q \in \varphi^{-1}(\tau(t)), \exists! t' \in \tilde{\varphi}^{-1}(t), \tau(t') = q. \end{aligned}$$

**Proposition 1.** *Let  $\mathcal{A}$  and  $\mathcal{B}$  be two weighted two-way automata. If  $\mathcal{A}$  is a covering (resp. an in-covering) of  $\mathcal{B}$ , the corresponding morphism  $\varphi$  induces a bijection between computations of  $\mathcal{A}$  and  $\mathcal{B}$  such that every computation of  $\mathcal{A}$  and its image in  $\mathcal{B}$  have the same label and the same weight.*

*Proof.* Assume that  $\mathcal{A}$  is a covering of  $\mathcal{B}$ . Let  $w$  be a word and let  $((p_0, i_0), \dots, (p_k, i_k))$  be a computation on  $w$  in  $\mathcal{A}$ . For every  $j$  in  $[0; k]$ , we set  $r_j = \varphi(p_j)$ ; by definition of a morphism  $((r_0, i_0), \dots, (r_k, i_k))$  is a computation on  $w$  in  $\mathcal{B}$  with the same weight. Conversely, let  $((r_0, i_0), \dots, (r_k, i_k))$  be a computation in  $\mathcal{B}$ . Let  $p_0$  be the unique initial state in  $\varphi^{-1}(r_0)$ . For every  $j$  in  $[0; k-1]$ , let  $\delta_j = i_{j+1} - i_j$ ; the configuration  $(r_{j+1}, r_{j+1})$  is reached from configuration  $(r_j, i_j)$  through the transition  $(r_j, w_j, \delta_j, r_{j+1})$ ; inductively, we define  $p_{j+1}$  as the unique state in  $\varphi^{-1}(r_{j+1})$  such that  $(p_j, w_j, \delta_j, p_{j+1})$  is a transition of  $\mathcal{A}$ . Then,  $((p_0, i_0), \dots, (p_k, i_k))$  is a computation on  $w$  in  $\mathcal{A}$ . Hence, every computation  $\rho$  of  $\mathcal{B}$  is lift up in a unique way into a computation of  $\mathcal{A}$  whose image by  $\varphi$  is  $\rho$ .

The proof is similar for in-coverings.  $\square$

This proposition implies that a two-way automaton and its covering (resp. in-covering) are equivalent; moreover, if a two-way automaton is unambiguous, so is every of its (in-)coverings.

### 2.3 $\delta$ -Locality

**Definition 6.** *Let  $\mathcal{A}$  be a two-way  $\mathbb{K}$ -automaton. If, for each state  $p$  of  $\mathcal{A}$ , every transition outgoing from  $p$  has the same direction, then  $\mathcal{A}$  is  $\delta$ -local.*

If  $Q$  is the set of states of a two-way  $\mathbb{K}$ -automaton, we denote  $Q_+$  (resp.  $Q_-$ ) the set of states  $p$  such that, for every transition  $t$  outgoing from  $p$ ,  $\delta(t) = +1$  (resp.  $\delta(t) = -1$ ); by convention, if  $p$  has no outgoing transition,  $p$  is in  $Q_+$ . For every state  $p$  of  $Q_+$  (resp.  $Q_-$ ), we set  $\delta(p) = 1$  (resp.  $\delta(p) = -1$ ).

If  $\mathcal{A}$  is a  $\delta$ -local automaton,  $\{Q_+, Q_-\}$  is a partition of  $Q$ .

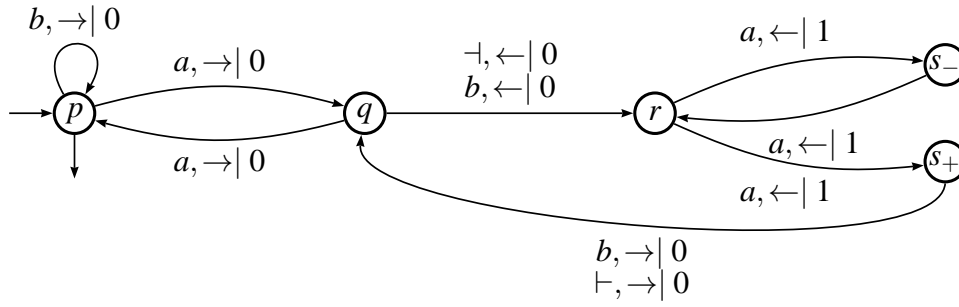
**Proposition 2.** *Every two-way  $\mathbb{K}$ -automaton admits a  $\delta$ -local in-covering.*

*Proof.* In this proof, we denote  $\pm = \{-1, +1\}$ . Let  $\mathcal{A} = (R, A, F, J, U)$  be a two-way  $\mathbb{K}$ -automaton and let  $P = R \setminus (R_+ \cup R_-)$  be the set of states in  $\mathcal{A}$  such that there are at least two transitions with different direction outgoing from each state. Let  $P_+$  and  $P_-$  be two copies of  $P$  and let  $Q = R_+ \cup R_- \cup P_+ \cup P_-$ . Let  $\varphi$  be the canonical mapping from  $Q$  onto  $R$ : it maps every element of  $P_+$  or  $P_-$  onto the corresponding element of  $P$ . Let  $\tilde{\varphi}$  be the mapping from  $Q \times A_{\pm} \times \pm \times Q$  into  $R \times A_{\pm} \times \pm \times R$  defined by  $\tilde{\varphi}(p, a, d, q) = (\varphi(p), a, d, \varphi(q))$ .

Let  $\mathcal{A}' = (Q, A, E, I, T)$  be the automaton defined by:

$$\begin{aligned} \underline{I} &= \varphi^{-1}(\underline{J}); \quad \underline{T} = \varphi^{-1}(\underline{U}) \setminus P_-; \\ \underline{E} &= \{(p, a, d, q) \in \tilde{\varphi}^{-1}(\underline{E}) \mid (p, d) \in (P_+ \cup R_+) \times \{+1\} \cup (P_- \cup R_-) \times \{-1\}\}; \\ \forall p \in \underline{I}, I(p) &= J(\varphi(p)), \quad \forall p \in \underline{T}, T(p) = U(\varphi(p)), \quad \forall t \in \underline{E}, E(t) = F(\tilde{\varphi}(p)). \end{aligned}$$

The automaton  $\mathcal{A}'$  is  $\delta$ -local and it is an in-covering of  $\mathcal{A}$ .  $\square$

Figure 3: The  $\delta$ -local two-way distance automaton  $\mathcal{A}'_1$ .

**Example 3.** The automaton  $\mathcal{A}_1$  of Figure 1 is not  $\delta$ -local; from state  $q$  (resp.  $s$ ), there are transitions leaving with  $\delta = 1$  and other ones with  $\delta = -1$ . The automaton  $\mathcal{A}'_1$  of Figure 3 is a  $\delta$ -local in-covering of  $\mathcal{A}_1$ . Notice that an in-covering of a deterministic automaton is not necessarily deterministic.

Actually, on  $\mathcal{A}'_1$ , transitions  $s_+ \xrightarrow{b, \rightarrow | 0} q_-$  and  $s_+ \xrightarrow{t, \rightarrow | 0} q_-$  do not belong to any computation, since the label of any transition that would follow one of these boths transitions should be the same as the label of the transition arriving at  $s_-$  ( $a$ ), and there is no transition outgoing from  $q_-$  with label  $a$ .

### 3 Slices

In this section, we describe the conversion of two-way automata over commutative semirings into one-way automata. We give sufficient conditions to get finite one-way automata.

#### 3.1 The Slice Automaton

**Definition 7.** Let  $\mathcal{A} = (Q, A, E, I, T)$  be a two-way  $\mathbb{K}$ -automaton and let  $w = w_1 \dots w_k$  be a word.  $\rho = ((p_0, i_0), \dots, (p_n, i_n))$  be a run over  $w$ , and  $j$  in  $[1; k+1]$ . Let  $h$  be the subsequence of all pairs  $(p_r, i_r)$  such that  $(i_r, i_{r+1}) = (j, j+1)$  or  $(i_{r-1}, i_r) = (j, j-1)$ . The  $j$ -th slice of  $\rho$  is the vector  $s^{(j)}$  of states obtained by the projection of the first component of each pair of  $h$ .

The signature  $S(\rho)$  of  $\rho$  is the sequence of its slices.

The slices we define here are not exactly the *crossing sequences* defined in [14].

**Example 4.** The vector  $\begin{bmatrix} q \\ r \\ p \end{bmatrix}$  is the second (and the seventh) slice of the run of Figure 2. The signature of this run is:

$$\left( \begin{array}{cc} p & q \\ s & r \\ q & p \end{array}, p, q, p, \begin{array}{cc} p & q \\ s & r \\ q & p \end{array} \right). \quad (1)$$

The signature of the (unique) run on the word  $abaaba$  in the automaton  $\mathcal{A}'_1$  is

$$\left( \begin{array}{cc} p & q_- \\ s_+ & r \\ q_+ & p \end{array}, p, q_+, p, \begin{array}{cc} p & q_- \\ s_+ & r \\ q_+ & p \end{array} \right). \quad (2)$$

Let  $\mathcal{A} = (Q, A, E, I, T)$  be a  $\delta$ -local two-way  $\mathbb{K}$ -automaton. To define a one-way  $\mathbb{K}$ -automaton from slices we consider the set  $X$  of subvectors of slices, that are vectors  $v$  in  $Q^*$  with an odd length; let  $Y$  be the vectors  $v$  in  $Q^*$  with an even length.

We define inductively two partial functions  $\theta : X \times A \times X \rightarrow \mathbb{K}$  and  $\eta : Y \times A \times Y \rightarrow \mathbb{K}$  by:

$$\begin{aligned} \eta(\varepsilon, a, \varepsilon) &= 0_{\mathbb{K}}, \\ \forall p, q \in Q, \quad \delta(p) = 1 &\implies \forall u, v \in Y, \theta(pu, a, qv) = E(p, a, 1, q) + \eta(u, a, v), \\ &\eta(u, a, pqv) = E(p, a, 1, q) + \eta(u, a, v), \\ \delta(p) = -1 &\implies \forall u, v \in X, \theta(pqu, a, v) = E(p, a, -1, q) + \theta(u, a, v), \\ &\eta(qu, a, pv) = E(p, a, -1, q) + \theta(u, a, yv). \end{aligned} \quad (3)$$

Since  $\mathcal{A}$  is  $\delta$ -local, for every triple  $(u, a, v)$  in  $X \times A \times X$ , if  $\theta(u, a, v)$  is defined, it is uniquely defined.

For every vector  $pu$  in  $X$ ,  $pu$  is initial if  $p$  is in  $\underline{I}$  and  $(\varepsilon, \vdash, u)$  is in  $\underline{\eta}$ ; in this case, we set  $\mathcal{I}(pu) = I(p) + \eta(\varepsilon, \vdash, u)$ . Likewise, every vector  $up$  in  $X$  is final if  $p$  is in  $\underline{T}$  and  $(u, \dashv, \varepsilon)$  is in  $\underline{\eta}$ ; in this case, we set  $\mathcal{T}(up) = \eta(u, \dashv, \varepsilon) + T(p)$ .

**Example 5.** For instance, with slices from automaton  $\mathcal{A}_1$ ,

$$\begin{aligned} \theta \left( \begin{array}{c} p \\ s_+ \\ q_+ \end{array}, a, \begin{array}{c} q_- \\ r \\ p \end{array} \right) &= E(p, a, 1, q_-) + \eta \left( \begin{array}{c} s_+ \\ q_+ \end{array}, a, \begin{array}{c} r \\ p \end{array} \right) \\ &= E(p, a, 1, q_-) + E(r, a, -1, s_+) + \theta(q_+, a, p) \\ &= E(p, a, 1, q_-) + E(r, a, -1, s_+) + E(q_+, a, 1, p). \end{aligned} \quad (4)$$

The vector  $\begin{bmatrix} p \\ s_+ \\ q_+ \end{bmatrix}$  is initial and

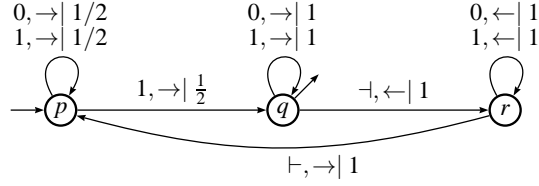
$$\mathcal{I} \left( \begin{array}{c} p \\ s_+ \\ q_+ \end{array} \right) = I(p) + E(s_+, \vdash, 1, q_+). \quad (5)$$

**Definition 8.** Let  $\mathcal{A} = (Q, A, E, I, T)$  be a two-way  $\mathbb{K}$ -automaton. With the above notations, the slice automaton of  $\mathcal{A}$  is the infinite one-way  $\mathbb{K}$ -automaton  $\mathcal{C} = (X, A, \theta, \mathcal{I}, \mathcal{T})$ .

**Proposition 3.** Let  $\mathbb{K}$  be a commutative semiring and let  $\mathcal{A}$  be a  $\delta$ -local two-way  $\mathbb{K}$ -automaton. There is a bijection  $\varphi$  between the computations of  $\mathcal{A}$  and the computations of the slice automaton of  $\mathcal{A}$  such that, for every computation  $\rho$  of  $\mathcal{A}$ ,

- $\rho$  and  $\varphi(\rho)$  have the same label and the same weight;
- the signature of  $\rho$  is the sequence of states of  $\varphi(\rho)$ .

*Proof.* Let  $\mathcal{C}$  be the slice automaton of  $\mathcal{A}$ . Let  $\pi$  be a run in  $\mathcal{C}$  with label  $w$ . Let  $\pi^{(k)}$  be the prefix of length  $k$  of  $\pi$  and let  $(v^{(0)}, \dots, v^{(k)})$  be the sequence of states of  $\pi^{(k)}$ . We show by induction on  $k$  that from  $\pi^{(k)}$ , there is a unique way to retrieve the restriction of a run of  $\mathcal{A}$  on  $w$  to the  $k$  first letters. Moreover, the weight of  $\pi^{(k)}$  (including initial weight) is equal to the weight of this restriction. If  $k = 0$ ,  $\pi^{(k)}$  is reduced to an initial slice. By Equation 3, the restriction of the path in the two-way automaton is uniquely defined:  $v_1^{(0)}$  is initial with weight  $I(v_1^{(0)})$ , and for every  $r$  in  $[1; (v^{(0)} - 1)/2]$ , there is a transition  $v_{2r}^{(0)} \xrightarrow{\vdash, \rightarrow | h_r} v_{2r+1}^{(0)}$ ; the weight of this restriction is actually the initial weight of  $v^{(0)}$  in  $\mathcal{C}$ . If  $k > 0$ , we consider the restriction built for  $k - 1$ ; this restriction corresponds to a disjoint union of parts of the

Figure 4: The two-way  $\mathbb{Q}$ -automaton  $\mathcal{A}_2$ .

computations and there is only one way to connect them to the states of the slice  $v^{(k)}$  (since  $\mathcal{A}$  is  $\delta$ -local). The weight of the transition between  $v^{(k-1)}$  and  $v^{(k)}$  is exactly the sum of the weights of the new transitions involved in the restriction.

Finally, from the restriction of length  $|w|$ , if we consider  $v^{(|w|)}$  as a final state of  $\mathcal{C}$ , by an argument similar to the initial state, we obtain that there is one and only one run in  $\mathcal{A}$  that corresponds to a given run in  $\mathcal{C}$ .  $\square$

### 3.2 Reduced computations and one-way automata

In unweighted (or Boolean) automata, two-way automata describe exactly the same languages as one-way automata [14, 12]. It is not always the case with weighted automata. For instance, let  $\mathbb{K}$  be the semiring of languages of the alphabet  $\{x, y\}$ . It is not difficult to design a deterministic two-way  $\mathbb{K}$ -automaton over the alphabet  $\{a\}$  such that the image of  $a^n$  is  $x^n y^n$  (a first left-right traversal outputs an  $x$  for each  $a$ , then the automaton comes back to the beginning of the word and a second left-right traversal outputs a  $y$  for each  $a$ ). This function is obviously not rational and can not be realized by a one-way  $\mathbb{K}$ -automaton.

**Proposition 4.** *Let  $\mathbb{K}$  be a commutative semiring and let  $\mathcal{A}$  be a  $\delta$ -local two-way  $\mathbb{K}$ -automaton. There exists a (finite) one-way  $\mathbb{K}$ -automaton  $\mathcal{B}$  such that there is a bijection  $\varphi$  between the reduced computations of  $\mathcal{A}$  and the computations of  $\mathcal{B}$  such that, for every reduced computation  $\rho$  of  $\mathcal{A}$ ,*

- $\rho$  and  $\varphi(\rho)$  have the same label and the same weight;
- the signature of  $\rho$  is the sequence of states of  $\varphi(\rho)$ .

*Proof.* Let  $\mathcal{A} = (Q, A, E, I, T)$  be a two-way  $\mathbb{K}$ -automaton. We consider vectors of elements of  $\mathbb{K}$  such that no state of  $Q$  appears twice at positions with the same parity. For all  $k$  in  $\mathbb{N}$ , we set

$$\begin{aligned} V_k &= \{v \in \mathbb{K}^{2k+1} \mid v_i = v_j \Rightarrow i \not\equiv j \pmod{2}\} \\ &= \{v \in \mathbb{K}^{2k+1} \mid \forall p \in Q, \forall s \in [0; 1], |\{i \mid v_i = p \text{ and } i \equiv s \pmod{2}\}| \leq 1\} \end{aligned} \quad (6)$$

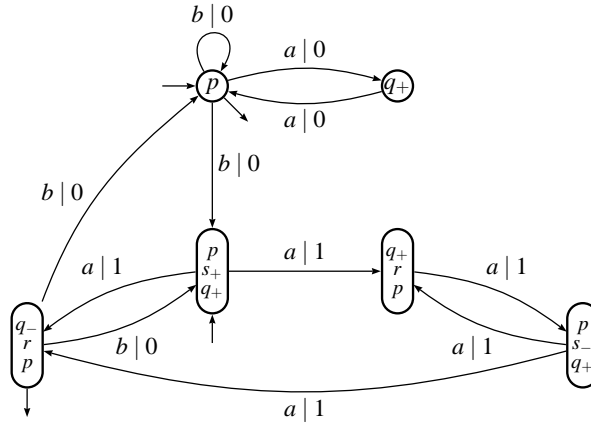
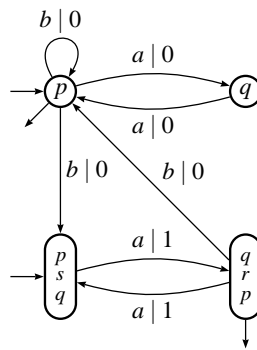
For every  $k$  larger than  $|Q| - 1$ ,  $V_k$  is empty. Let  $V = \bigcup_k V_k$ ; we define the one-way  $\mathbb{K}$ -automaton with set of states  $V$ . It is straightforward that a run is reduced if and only if every slice of this run is in  $V$ .

By Proposition 3, the restriction of the slice automaton to  $V$  gives a finite automaton that fulfils the proposition.  $\square$

Actually, the sufficient condition for the finiteness of the trim part of the slice automaton can be weakened. If the number of slices of a two-way automaton is finite, it is equivalent to a one-way automaton. Unfortunately, this condition is not easy to check and is not a necessary condition.

**Example 6.** *The two-way  $\mathbb{Q}$ -automaton  $\mathcal{A}_2$  computes for each word  $w$  over the alphabet  $\{0, 1\}$  the value  $\frac{x}{1-x}$ , where  $x = \sum_{i \in [1; |w|]} \frac{w_i}{2^i}$ . Although  $\mathbb{Q}$  is commutative, this two-way automaton is not equivalent to any one-way  $\mathbb{Q}$ -automaton;  $s_2 = |\mathcal{A}_2|$  is not a rational series.*



Figure 5: The unambiguous one-way distance automata  $\mathcal{B}_1$ .Figure 6: The unambiguous one-way distance automata  $\mathcal{B}_1$ .

**Example 7.** Let  $\mathcal{B}'_1$  be the trim part of the slice automaton of  $\mathcal{A}'_1$  (Figure 5). In this particular case, although  $\mathcal{A}_1$  is not  $\delta$ -local, the slice automaton  $\mathcal{B}_1$  of  $\mathcal{A}_1$  (Figure 6) is also unambiguous. It has been shown in [8] that there is no deterministic one-way distance automaton equivalent to these automata.

## 4 Unambiguity and Determinism

Since every computation in an unambiguous two-way automaton is reduced, Proposition 4 implies the following statement.

**Proposition 5.** Let  $\mathbb{K}$  be a commutative semiring. Every unambiguous two-way  $\mathbb{K}$ -automaton is equivalent to an unambiguous one-way  $\mathbb{K}$ -automaton.

A unambiguous one-way automaton can obviously be seen as a unambiguous two-way automaton. In this part, we show that an unambiguous one-way automaton can actually be converted into a deterministic two-way automaton.

### 4.1 From Unambiguous one-way to Deterministic two-way Automata

**Definition 9.** A two-way automaton is deterministic if  
i) it has at most one initial state;

- ii) for every state  $p$  and every letter  $a$ , there is at most one transition outgoing from  $p$  with label  $a$ ;  
 iii) for every final state  $p$ , there is no transition outgoing from  $p$  with label  $\neg$ .

The last condition means that if a final state is reached at the end of the word, there is no nondeterministic choice between ending the computation and reading the right mark to continue.

**Theorem 1.** *Let  $\mathbb{K}$  be a semiring. Every unambiguous one-way  $\mathbb{K}$ -automaton is equivalent to a deterministic two-way  $\mathbb{K}$ -automaton.*

This result is an extension of [7], where it is proved that an unambiguous one-way automaton can be simulated by a deterministic two-way automaton. Our proof is inspired by [3], where it is proven that any rational function can be realized by a sequential two-way transducer. Other works on the conversion of two-way transducers to one-way transducers can be found in [5] or in [4].

*Proof.* Let  $\mathcal{A} = (I, E, T)$  an unambiguous one-way  $\mathbb{K}$ -automaton with set of states  $Q$ .

We consider the mapping  $\mu$  from  $A$  into the  $Q \times Q$  Boolean matrices defined by:

$$\forall a \in A, \forall p, q \in Q, \mu(a)_{p,q} = 1 \iff (p, a, q) \in \underline{E}. \quad (7)$$

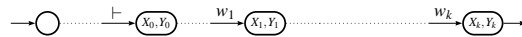
The monoid generated by  $\{\mu(a) \mid a \in A\}$  is the *transition monoid*  $M$  of  $\mathcal{A}$ . The mapping  $\mu$  is naturally extended to a morphism of the monoid  $A^*$  onto  $M$ . Every subset of  $Q$  can be interpreted as a vector in  $\mathbb{B}^Q$ ; for every word  $w$ ,  $\underline{I}\mu(w)$  is the set of states accessible from an initial state by a path with label  $w$  and conversely,  $\mu(w)\underline{T}$  is the set of states from which a terminal state can be reached by a path with label  $w$ .

Since  $\mathcal{A}$  is unambiguous, for every pair of words  $(u, v)$ ,  $\underline{I}\mu(u) \cap \mu(v)\underline{T}$  has at most one element (otherwise there would exist several computations accepting  $uv$ ); likewise, for every letter, there exists at most one transition  $(p, a, q)$  in  $\mathcal{A}$  with  $p$  in  $\underline{I}\mu(u)$  and  $q$  in  $\mu(v)\underline{T}$  (otherwise there would exist several computations accepting  $uav$ ).

For every word  $w = w_1 \dots w_k$ , for every  $i$  in  $[0; k]$ , we set

$$X_i(w) = \underline{I}\mu(w_1 \dots w_i) \quad \text{and} \quad Y_i(w) = \mu(w_{i+1} \dots w_k)\underline{T}.$$

We build a deterministic two-way  $\mathbb{K}$ -automaton  $\mathcal{B}$  equivalent to  $\mathcal{A}$ .  $\mathcal{B}$  has the following property. If  $w$  is accepted by  $\mathcal{B}$ , for every  $i$  in  $[1; k]$ , the state reached after the last reading of  $w_i$  contains the information  $(X_i(w), Y_i(w))$ :



From  $(X_{i-1}(w), Y_{i-1}(w))$  and  $(X_i(w), Y_i(w))$ , the transition labeled by  $w_i$  in the run with label  $w$  can be deduced: it is the only transition  $(p, w_i, q)$  with  $p$  in  $X_{i-1}(w)$  and  $q$  in  $Y_i(w)$ . Likewise  $(X_0, Y_0)$  determines the initial weight and  $(X_k, Y_k)$  determines the final weight.

The set  $X_i$  can easily be deduced from  $X_{i-1}$ :  $X_i = X_{i-1}\mu(w_i)$ . the computation of  $Y_i$  from  $Y_{i-1}$  is more subtle.

Let  $x$  and  $y$  be two elements of  $M$ . If there exists  $z$  in  $M$  such that  $x = zy$ , we say that  $x \leq_L y$ ; this relation is a preorder. If there also exists  $t$  such that  $tx = y$ , we say that  $x$  and  $y$  are L-equivalent.

Let  $u$  be a factor of  $w$  that starts in  $w_{i+1}$ . It obviously holds  $\mu(w_i u) \leq_L \mu(u)$ . If  $\mu(w_i u)$  and  $\mu(u)$  are L-equivalent, there exists  $y$  in  $M$  such that  $y\mu(w_i u) = \mu(u)$ . In this case, it also holds  $y\mu(w_i \dots w_k) = \mu(w_{i+1} \dots w_k)$  and therefore,  $Y_i = yY_{i-1}$ . The two-way automaton can perform these computations, since they lie in the transition monoid, which is finite. The automaton incrementally computes for each  $j$  in  $[i; k]$  the value of  $\mu(w_{i+1} \dots w_j)$  until  $\mu(w_i \dots w_j) <_L \mu(w_{i+1} \dots w_j)$  and  $\mu(w_i \dots w_{j+1}) \equiv_L \mu(w_{i+1} \dots w_{j+1})$ . If

it reaches  $j = k$ , then  $Y_i = \mu(w_{i+1} \dots w_j)T$ , otherwise,  $Y_i = yY_{i-1}$  where  $y$  is such that  $y\mu(w_i \dots w_{j+1}) = \mu(w_{i+1} \dots w_{j+1})$ .

Once  $Y_i$  is computed, the automaton must come back to position  $i$ . The automaton is in some position  $j$  such that  $\mu(w_i \dots w_j) <_L \mu(w_{i+1} \dots w_j)$ ; *a fortiori*, for every  $r$  in  $[i+1; j]$ ,  $\mu(w_i \dots w_j) <_L \mu(w_r \dots w_j)$ . The automaton therefore spans every position smaller than  $j$  until it arrives to some point  $s$  such that  $\mu(w_i \dots w_j) = \mu(w_s \dots w_j)$ . It then holds  $s = i$ .

Let  $\mathcal{P}$  be the powerset of  $\mathcal{Q}$ . The set of states of  $\mathcal{B}$  is the union of five kinds of states:

- $Q_0 = \{i\}$  is the initial state; in this state, the automaton read the input from left to right until it reached the right mark  $\dashv$ . It then goes to the state  $\underline{T}$  in  $Q_1$ .
- $Q_1 \subseteq \mathcal{P}$ ; in this state, the automaton read the input  $w$  from right to left; after reading the suffix  $v$ , the state corresponds to  $\mu(v)\underline{T}$ . When the left mark  $\vdash$  is reached, the automaton goes to the state  $(\underline{I}, \mu(w)\underline{T})$  in  $Q_2$ .
- $Q_2 \subseteq \mathcal{P}^2$ ; these states corresponds to the pairs  $(X_i, Y_i)$ ; the incoming transitions on these states correspond to the transition of the one-way automaton; they are weighted by the corresponding weight. Likewise, a state in  $Q_2$  may be terminal if it belongs to  $\mathcal{P} \times \{T\}$ . When the automaton is in one of these states, either it stops, or it starts to deal with a new letter; this letter is read and stored in the next state which belongs to  $Q_3$ .
- $Q_3 \subseteq A \times M \times \mathcal{P}^2$ ; the automaton stays in states  $Q_3$  as long as it needs to compute  $Y_i$  from  $Y_{i-1}$ . It stores the current letter  $a$  as well as the image in the transition monoid of the factor  $u$  that follows  $a$  and ends at the current position. If the state stores  $\mu(u)$  that is  $L$ -larger than  $\mu(au)$  and the read letter  $b$  is such that  $\mu(aub)$  and  $\mu(ub)$  are  $L$ -equivalent, there exists  $y$  such that  $y\mu(aub) = \mu(ub)$ ; then  $Y_i = yY_{i-1}$ , the automaton stores  $\mu(au)$  and jump to a state in  $Q_4$ .
- $Q_4 \subseteq M^2 \times \mathcal{P}^2$ ; the automaton stays in a state of  $Q_4$  while it reads from right to left the word  $u$ ; it stores the image of the suffix  $v$  of  $u$  which is read; it holds  $\mu(v) >_L \mu(au)$  until  $v = u$ ; at this point, the automaton read the letter  $a$  and checks that  $\mu(av) = \mu(au)$ ; at this point, it knows both  $X_i$  and  $Y_{i+1}$ , therefore, it can output the weight of the unique transition compatible with  $a$ ,  $X_i$  and  $Y_{i+1}$ , and jump to the state  $(X_{i+1} = X_i\mu(a), Y_{i+1})$ .

$$\begin{aligned}
F = & \{i \xrightarrow{a, \rightarrow} i \in Q_0 \mid a \in A\} \\
& \cup \{i \xrightarrow{\dashv, \leftarrow} \underline{T} \in Q_1\} \\
& \cup \{Y \xrightarrow{a, \leftarrow} \mu(a)Y \in Q_1 \mid Y \in Q_1, a \in A\} \\
& \cup \{Y \xrightarrow{\vdash, \rightarrow | I_k} (I, Y) \in Q_2 \mid Y \in Q_1, k \in I \cap Y\} \\
& \cup \{(X, Y) \xrightarrow{a, \rightarrow} (a, 1_{\mathbb{K}}, X, Y) \in Q_3 \mid (X, Y) \in Q_2, a \in A\} \\
& \cup \{(a, x, X, Y) \xrightarrow{b, \rightarrow} (a, x\mu(b), X, Y) \in Q_3 \mid (a, x, X, Y) \in Q_3, b \in A, \mu(a)x\mu(b) <_L x\mu(b)\} \\
& \cup \{(a, x, X, Y) \xrightarrow{b, \leftarrow} (\mu(a)x, 1, X, yY) \in Q_4 \mid (a, x, X, Y) \in Q_3, b \in A, y \in M, y\mu(a)x\mu(b) = x\mu(b)\} \\
& \cup \{(a, x, X, Y) \xrightarrow{\dashv, \leftarrow} (\mu(a)x, 1, X, T) \in Q_4 \mid (a, x, X, Y) \in Q_3\} \\
& \cup \{(x, y, X, Y) \xrightarrow{a, \leftarrow} (x, \mu(a)y, X, Y) \in Q_4 \mid (x, y, X, Y) \in Q_4, a \in A, x <_L \mu(a)y\} \\
& \cup \{(x, y, X, Y) \xrightarrow{a, \rightarrow | k} (X\mu(a), Y) \in Q_2 \mid (x, y, X, Y) \in Q_4, a \in A, x = \mu(a)y, \\
& \qquad \qquad \qquad \exists (p, q) \in X \times Y, \exists p \xrightarrow{a | k} q \in \mathcal{A}\}.
\end{aligned}$$

□

For every  $X$  in  $\mathcal{P}$ , if the state  $(X, \underline{T})$  belongs to  $Q_2$ ,  $(X, \underline{T})$  is final with weight  $T_p$ , where  $p$  is the unique state in  $X \cap \underline{T}$ .

**Example 8.** Let  $\mathcal{B}_1$  be the unambiguous one-way automaton of Figure 6. We number the states of this automaton:  $[p] = 1$ ,  $[q] = 2$ ,  $[p, s, q] = 3$  and  $[q, r, p] = 4$ . The transition monoid is generated by the following matrices:

$$\alpha = \mu(a) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \beta = \mu(b) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}. \quad (8)$$

The following identities hold :  $\alpha^2 = 1$ ,  $\beta^2 = \beta$ ,  $\beta\alpha\beta = \beta$ . It then holds  $1 \equiv_L \alpha$ ,  $\alpha\beta \equiv_L \beta$  and  $\alpha\beta\alpha \equiv_L \beta\alpha$ , while  $\beta <_L 1$  and  $\beta\alpha <_L 1$ . Notice that  $\beta$  and  $\beta\alpha$  are uncomparable. We can apply the proof of Theorem 1 to compute the equivalent deterministic two-way automaton  $\mathcal{D}_1$  of Figure 7.

**Corollary 1.** Let  $\mathbb{K}$  be a commutative semiring. Every unambiguous two-way  $\mathbb{K}$ -automaton is equivalent to a deterministic one.

**Remark 1.** This conversion can lead to a combinatorial blow-up. For instance, the deterministic two-way automaton built from the unambiguous one-way automato  $\mathcal{B}_1$  (Figure 6 (right)) has 27 states in its trim part.

A lower bound on the number of states can be computed. Let  $n$  be the number of states of the unambiguous one-way automaton.

- $Q_0$  has one state;
- $Q_1$  has at most  $2^n - 1$  states;
- $Q_2$  is made of pairs of subset of  $Q$  which share exactly one element, hence  $Q_2$  has at most  $n3^{n-1}$  states;
- $Q_3$  is made of a pair of  $Q_2$  endowed with a letter and an element of the transition monoid (that may have  $2^{n^2}$  elements); hence  $Q_3$  has at most  $|A|n3^{n-1}2^{n^2}$  states;
- $Q_4$  is made of two (non empty) subsets of  $Q$  and two elements of the transition monoid; its size is bounded by  $2^{2n+2n^2}$ .

## References

- [1] Marcella Anselmo (1990): *Two-way Automata with Multiplicity*. In: *ICALP'90, Lect. Notes in Comput. Sci.* 443, pp. 88–102, doi:10.1007/BFb0032024.
- [2] Marie-Pierre Béal, Olivier Carton, Christophe Prieur & Jacques Sakarovitch (2003): *Squaring transducers: an efficient procedure for deciding functionality and sequentiality*. *Theor. Comput. Sci.* 292(1), pp. 45–63, doi:10.1016/S0304-3975(01)00214-6.
- [3] Olivier Carton (2012): *Two-Way Transducers with a Two-Way Output Tape*. In: *DLT'12, Lect. Notes in Comput. Sci.* 7410, pp. 263–272, doi:10.1007/978-3-642-31653-1\_24.
- [4] Rodrigo De Souza (2013): *Uniformisation of Two-Way Transducers*. In: *LATA'13, Lect. Notes in Comput. Sci.* 7810, pp. 547–558, doi:10.1007/978-3-642-37064-9\_48.

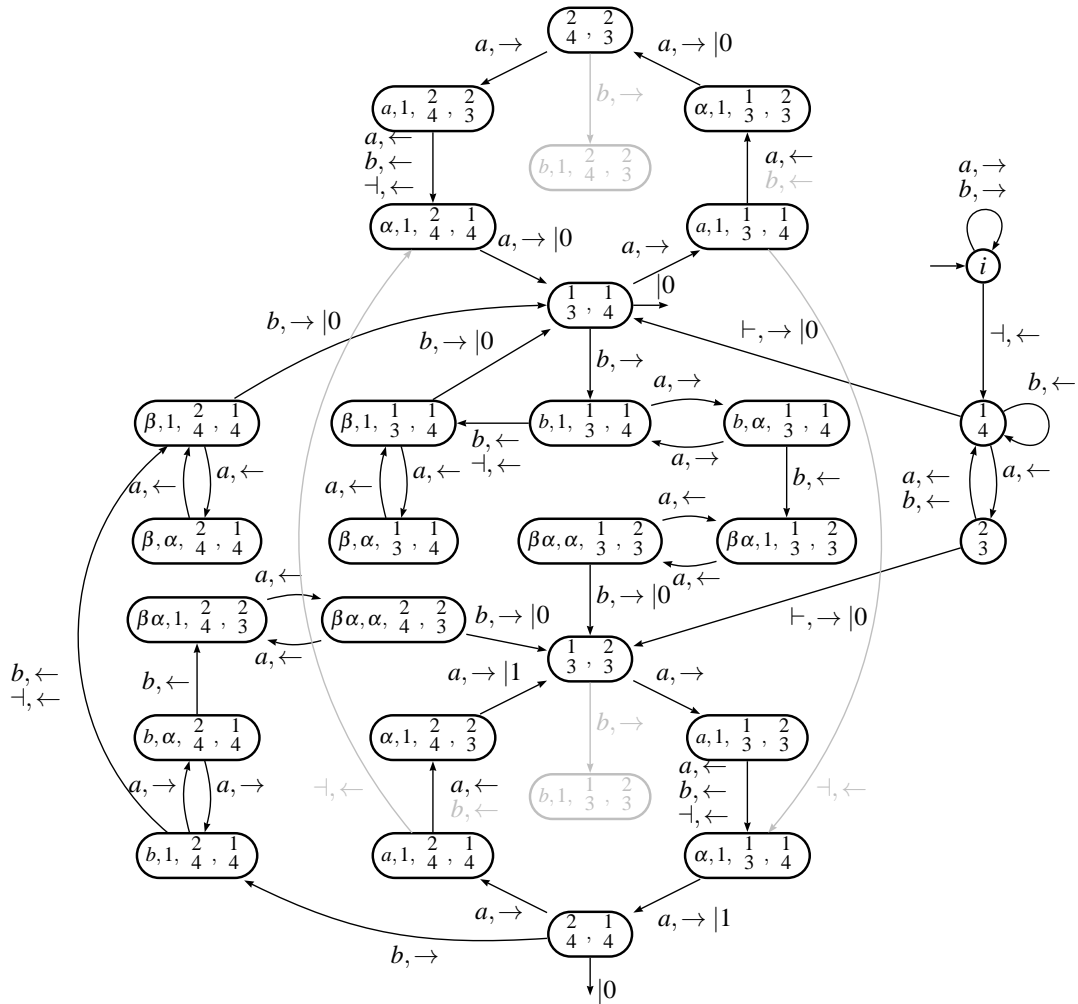


Figure 7: The deterministic two-way distance automaton  $\mathcal{D}_1$ . The transitions or states in gray are not accessible. For sake of clearness, transitions outgoing from non accessible states are not drawn. Every column of numbers is the set of non-zero components of a Boolean vector of size 4. The weights are only written on transitions where it comes from the weight of a transition (or from an initial/final weight) of  $\mathcal{B}_1$ .

- [5] Joost Engelfriet & Hendrik Jan Hoogetboom (2007): *Finitary Compositions of Two-way Finite-State Transductions*. *Fundam. Inform.* 80(1-3), pp. 111–123. Available at <http://iospress.metapress.com/content/143422w0253h8644/>.
- [6] Zoltán Ésik & Werner Kuich (2009): *Handbook of Weighted Automata*, chapter Finite Automata, pp. 69–104. Springer, doi:10.1007/978-3-642-01492-5.
- [7] John E. Hopcroft & Jeffrey D. Ullman (1967): *An Approach to a Unified Theory of Automata*. In: SWAT (FOCS), IEEE Computer Society, pp. 140–147, doi:10.1109/FOCS.1967.4.
- [8] Ines Klimann, Sylvain Lombardy, Jean Mairesse & Christophe Prieur (2004): *Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton*. *Theor. Comput. Sci.* 327(3), pp. 349–373, doi:10.1016/j.tcs.2004.02.049.
- [9] Daniel Kroh (1994): *The equality problem for rational series with multiplicities in the tropical semiring is undecidable*. *Internat. J. Algebra Comput.* 4(3), pp. 405–425, doi:10.1142/S0218196794000063.
- [10] Sylvain Lombardy & Jean Mairesse (2006): *Series which are both max-plus and min-plus rational are unambiguous*. *RAIRO - Theor. Inf. and Appl.* 40(1), pp. 1–14, doi:10.1051/ita:2005042.
- [11] Sylvain Lombardy & Jacques Sakarovitch (2013): *The validity of weighted automata*. *Internat. J. Algebra Comput.* 23, pp. 863–913, doi:10.1142/S0218196713400146.
- [12] M. O. Rabin & D. Scott (1959): *Finite automata and their decision problems*. *IBM J. Res. Dev.* 3(2), pp. 114–125, doi:10.1147/rd.32.0114.
- [13] Jacques Sakarovitch (2009): *Elements of Automata Theory*. Cambridge University Press, doi:10.1017/CBO9781139195218.
- [14] J. C. Shepherdson (1959): *The reduction of two-way automata to one-way automata*. *IBM J. Res. Dev.* 3(2), pp. 198–200, doi:10.1147/rd.32.0198.