City Research Online

City St George's, University of London

Citation: Daviaud, L. (2020). Containment and equivalence of weighted automata: Probabilistic and max-plus cases. In: Language and Automata Theory and Applications. LATA 2020. Lecture Notes in Computer Science, 12038. (pp. 17-32). Cham: Springer Verlag. ISBN 9783030406073 doi: 10.1007/978-3-030-40608-0_2

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/24184/

Link to published version: https://doi.org/10.1007/978-3-030-40608-0_2

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

Containment and Equivalence of Weighted Automata: Probabilistic and Max-Plus Cases

Laure Daviaud[0000−0002−9220−7118]

CitAI, Department of Computer Science, SMCSE, City University of London, UK
laure.daviaud@city.ac.uk

Abstract. This paper surveys some results regarding decision problems for probabilistic and max-plus automata, such as containment and equivalence. Probabilistic and max-plus automata are part of the general family of weighted automata, whose semantics are maps from words to real values. Given two weighted automata, the equivalence problem asks whether their semantics are the same, and the containment problem whether one is point-wise smaller than the other one. These problems have been studied intensively and this paper will review some techniques used to show (un)decidability and state a list of open questions that still remain.

Keywords: Weighted automata · Probabilistic automata · Max-plus automata · Equivalence problem · Containment problem · Decidability

1 Introduction

Weighted automata have been introduced by Schützenberger in 1961 in [37] as a quantitative generalisation of non deterministic finite-state automata. While non deterministic finite automata have a Boolean behaviour (each word is mapped to 0 or 1), weighted automata allow a more fine grained output: each word is mapped to an element in a chosen semiring. This allows for example to map words with real values, modelling probabilities or costs. They have been intensively studied both:

1. in a general setting, i.e. giving frameworks and results that are valid for any semiring,
2. and on particular instances, for example when focusing on the classic semiring $\mathbb{R}$ with the standard addition and product operations.

For any semiring, a weighted automaton can be viewed as a graph with labelled transitions carrying weights, or as a morphism from words to matrices (both definitions are given in section 2). Recently, Alur introduced another equivalent model, closer to the implementation level: the cost register automata [2, 3]. These have engendered a lot of research works, in particular regarding questions around minimisation. Weighted automata also admit an equivalent logic [14] introduced by Droste and Gastin and while this paper focuses on weighted

automata on words, they have also been generalised to other structures such as trees [4, 38, 29].

Specific instances that have been particularly studied are the probabilistic automata on the standard semiring $\mathbb{R}$ with operations $+$ and $\times$ (the exact definition is given in Section 2) [35]; and the max-plus (resp. min-plus) automata on the semiring $\mathbb{R}$ with operations max (resp. min) and $+$ [40].

They have been applied in image compression [25], natural language processing [6, 33, 32], model-checking of probabilistic systems [42, 30], automatic analysis of complexity of programs [9], study of discrete event systems [18], and in the proofs of results in tropical algebra [10] and automata theory [21, 39, 40].

One of the first natural question which arises when dealing with computational models is the equivalence problem: in our case, this would ask whether two distinct weighted automata map words to the same values? Since probabilistic and max-plus automata compute functions from words to real values, another natural problem is to wonder whether the function computed by a given probabilistic (resp. max-plus) automaton is point-wise smaller than the function computed by another probabilistic (resp. max-plus) automaton. This is called the containment problem. These problems are highly dependant on the semiring under consideration and have originally been tackled using very different techniques for probabilistic and max-plus automata. We will however present one technique that can be used in both cases to show the undecidability of the containment problem for both max-plus and probabilistic automata which are linearly ambiguous.

Another mainstream topic that have been intensively studied is the one of determinisation. Weighted automata are not determinisable in general: there exist for example max-plus automata that do not have an equivalent deterministic one. This question is of particular interest for max-plus automata and is linked to the minimisation of cost register automata [13, 11]. Deciding whether a given max-plus automaton is determinisable is still open. This topic is out of the scope of this paper but the interested reader is referred to [27, 26, 17].

In the rest of this paper, we will explain a way to prove undecidability of the containment problem for probabilistic and max-plus automata and discuss restricted classes as well as approximations to obtain more positive results and decidability in some cases.

2 Weighted automata

In this section, we start by recalling basic notions used to define weighted automata. The paper should be self-contained but the reader is referred to [36] for a full exposition on the topic.

2.1 Preliminaries

Semiring. Given a set M, a binary operation $\cdot$ on M and an element 1 of M (called neutral element), $(M, \cdot, 1)$ is called a monoid if $\cdot$ is associative and

$1 \cdot x = x \cdot 1 = x$ for all $x \in M$. The monoid is said to be commutative if $x \cdot y = y \cdot x$ for all $x, y \in M$.

A semiring $(S, \oplus, \otimes, 0, 1)$ is a set S equipped with two binary operations such that $(S, \oplus, 0)$ is a commutative monoid, $(S, \otimes, 1)$ is a monoid, $0 \otimes x = x \otimes 0 = 0$ and $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$ and $(y \oplus z) \otimes x = (y \otimes x) \oplus (z \otimes x)$ for all $x, y, z \in S$. In the rest of the paper, we will simply use S to denote the semiring $(S, \oplus, \otimes, 0, 1)$. For x in S, we will use the standard notation x^k to denote the product $\underbrace{x \otimes \cdots \otimes x}_{k \text{ times}}$.

Matrices. Given a semiring S, we consider the set of matrices with coefficients in S. Given a matrix M, $M[i][j]$ denotes the coefficient at row i and column j of the matrix. For two matrices M and M', one defines the product $M \cdot M'$ provided the number of columns of M is equal to the number of rows of M' (denoted by N) by:

$$(M \cdot M')[i][j] = \bigoplus_{k=1,\ldots,N} (M[i][k] \otimes M'[k][j])$$

Given a positive integer N, the set of square matrices of dimension $N \times N$ with coefficients in S is denoted by $\mathcal{M}_N(S)$ or simply $\mathcal{M}_N$ when S is clear from context. The set $\mathcal{M}_N$ equipped with the binary operation $\cdot$ is a monoid with neutral element the identity matrix (the matrix with 1 on the diagonal coefficients and 0 everywhere else).

Words. In the rest of the paper, Σ denotes a finite alphabet, Σ^* the set of words on this alphabet, and ε the empty word. For a word w, $|w|$ denotes the length of w.

Notation. Given a finite set Q, $|Q|$ denotes the number of elements in Q.

2.2 Graphical definition

We give now a first definition of weighted automata.

Definition 1. *A weighted automaton over a semiring S and alphabet Σ is a tuple (Q, Q_I, Q_F, T) where:*

- *Q is a finite set of states,*
- *$Q_I \subseteq Q$ is the set of initial states,*
- *$Q_F \subseteq Q$ is the set of final states,*
- *T is the transition function $Q \times \Sigma \times Q \to S$.*

Given $p, q \in Q$ and $a \in \Sigma$, whenever $T(p, a, q) \neq 0$, we say that (p, a, q) is a transition, $T(p, a, q)$ is its weight and we write:

$$p \xrightarrow{a:T(p,a,q)} q$$

A run on a word $w = w_1 w_2 \dots w_n$ where for all $i = 1, \dots, n$, $w_i \in \Sigma$ is a sequence of compatible transitions:

$$q_0 \xrightarrow{w_1:m_1} q_1 \xrightarrow{w_2:m_2} q_2 \xrightarrow{w_3:m_3} \cdots \xrightarrow{w_n:m_n} q_n$$

The weight of a run is the product of the weights of the transitions in the run i.e. $\bigotimes_{i=1}^{n} m_i$. A run is said to be accepting if $q_0 \in Q_I$ and $q_n \in Q_F$. The weight of a word in the automaton is the sum $\oplus$ of the weights of the accepting runs on w, and 0 if there is no such run. By convention, the weight of the empty word is 1 if there exist a state which is both initial and final and 0 otherwise.

Definition 2. *The semantics of a weighted automaton $\mathcal{A}$ over the semiring S and alphabet Σ is the function which maps every word of Σ^* to its weight in S. It is denoted by $[\![\mathcal{A}]\!]$.*

Variants. There exist several alternative definitions for weighted automata. A classic one allows also initial and final weights: each state q is associated with two elements i_q and f_q from S (possibly 0). The weight of a run:

$$q_0 \xrightarrow{w_1:m_1} q_1 \xrightarrow{w_2:m_2} q_2 \xrightarrow{w_3:m_3} \cdots \xrightarrow{w_n:m_n} q_n$$

is then the product $i_{q_0} \otimes m_1 \otimes \cdots \otimes m_n \otimes f_{q_n}$ and the weight of a word w the sum of the weights of the runs on w.

Adding initial and final weights does not increase the expressive power of weighted automata: the set of semantics is the same. The problems we are considering in this paper are also not affected by this: equivalence or containment will be decidable for both variants or for none.

However, these considerations will have to be taken into account when dealing with determinisation issues, but this is not in the scope of this paper.

Example 1. Figure 1 shows a weighted automaton on a semiring S and alphabet $\{a, b, t\}$. It has two initial states q_1 and q_2 and two final states q_3 and q_4. Let w be the word $a^{i_0} b^{j_0} t a^{i_1} b^{j_1} t \cdots t a^{i_k} b^{j_k}$ for some non negative integers $i_0, j_0, \dots$ There are k accepting runs on w: each accepting run corresponds to read one of the occurrences of t from q_2 to q_3 and has weight $m^{i_\ell} \otimes n^{i_{\ell+1}}$ for some $\ell \in \{0, \dots, k-1\}$. The weight of w is then:

$$\bigoplus_{\ell \in \{0,\dots,k-1\}} m^{i_\ell} \otimes n^{i_{\ell+1}}$$

Ambiguity. The notion of ambiguity that applies for finite non deterministic automata applies here too: A weighted automaton is said to be unambiguous is there is at most one accepting run on every word, and finitely ambiguous if there is an integer L such that for all words w, there are at most L accepting runs labelled by w. Finally, a run is polynomially ambiguous if there is a polynomial P such that for all words w, there are at most $P(|w|)$ accepting runs labelled by w, and linearly ambiguous whenever this polynomial is of degree 1.

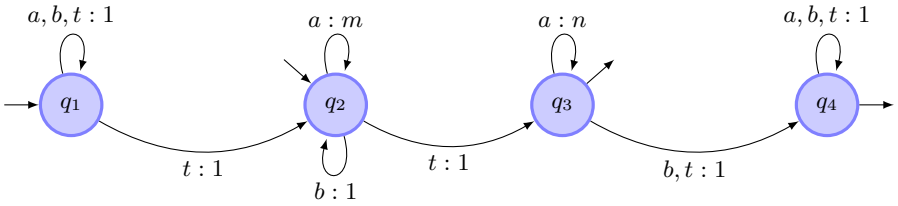


Fig. 1. Weighted automaton

2.3 Matrix representation

An equivalent way to see weighted automata is by using a matrix representation.

Definition 3 (equivalent definition). *A weighted automaton over a semiring S and alphabet Σ is a tuple (N, I, F, μ) where N is an integer and:*

- *I is a set of matrices with 1 row and N columns, and coefficients in $\{0, 1\}$,*
- *F is a set of matrices with N rows and 1 column, and coefficients in $\{0, 1\}$,*
- *μ is a map $\Sigma \to \mathcal{M}_N(S)$.*

The semantics of a weighted automaton is a function mapping each word $w = w_1 w_2 \cdots w_n$, where for all i, $w_i \in \Sigma$ to:

$$I \cdot \mu(w_1) \cdot \mu(w_2) \cdots \mu(w_n) \cdot F$$

Note that the later is an element of S. There are easy translations to go from the graphical definitions to the matrix one and conversely:

- starting from (Q, Q_I, Q_F, T), set $N = |Q|$, and denote by $q_1, q_2, \ldots, q_N$ the states in Q. Set I with 1 row and N columns, defined by $I[1][j] = 1$ if $q_j \in Q_I$ and 0 otherwise. Similarly, set F with N rows and 1 column, defined by $F[j][1] = 1$ if $q_j \in Q_F$ and 0 otherwise. Finally, set μ such that for all $a \in \Sigma$, $\mu(a)[i][j] = T(q_i, a, q_j)$. It is easy to check that the semantics of the two weighted automata are the same. In particular, for any word w, $\mu(w)[i][j]$ is the sum of the weights of the runs labelled by w going from q_i to q_j.
- Similarly, starting from (N, I, F, μ), set $Q = \{q_1, q_2, \ldots, q_N\}$, $Q_I = \{q_j \mid I[1][j] = 1\}$, $Q_F = \{q_j \mid F[j][1] = 1\}$ and $T(q_i, a, q_j) = \mu(a)[i][j]$ for all $a \in \Sigma$, $i, j \in \{1, \ldots, N\}$.

Example 2. The weighted automaton from Figure 1 is represented by the following set of matrices:

$$\mu(a) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & n & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mu(b) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mu(t) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$I = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix} \quad F = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

2.4 Classic examples

In the rest of the paper, we will consider two classic examples of weighted automata: max-plus automata and probabilistic automata.

Non deterministic finite automata (or Boolean automata). Non deterministic finite automata can be seen as weighted automata over the Boolean semiring. They can also be seen as weighted automata in any semiring as follows: for a given Boolean automaton $\mathcal{B}$, and a semiring S, construct the weighted automaton $\mathcal{A}$ by giving weight 1 to transitions in $\mathcal{B}$ (and 0 to non-existing transitions). It is easy to see that for every word w, w is accepted by $\mathcal{B}$ if and only if $[\![\mathcal{A}]\!](w) = 1$ ($[\![\mathcal{A}]\!](w) = 0$ otherwise).

Max-plus automata. The semiring $\{\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0\}$ is called the max-plus semiring. Max-plus automata are weighted automata over the max-plus semiring. Alternatively, the min-plus semiring is $\{\mathbb{R} \cup \{+\infty\}, \min, +, +\infty, 0\}$ and min-plus automata are weighted automata over this semiring. Given a max-plus automaton $\mathcal{A}$, it is easy to construct a min-plus automaton $\mathcal{B}$ by changing all the weights to their opposite to obtain $[\![\mathcal{A}]\!] = -[\![\mathcal{B}]\!]$. Most of the results later on given for max-plus automata can thus be easily translated for min-plus automata. However, when restricting the domain of the semiring to $\mathbb{N}$, the translation does not work anymore and for the results which are only valid in $\mathbb{N}$, one has to be more careful when translating from max-plus to min-plus and vice-versa.

Probabilistic automata. Probabilistic automata are weighted automata over the semiring $\{\mathbb{R}, +, \times, 0, 1\}$ with the extra restriction that all the weights on transitions are in $[0, 1]$ and for a given state q and a given letter a of Σ, the weights of the transitions exiting q labelled by a have to sum to 1.

Example 3. Let us consider the weighted automaton given in Figure 1 and the word w defined in Example 1. For S taken as the max-plus semiring, the weight of w would be:

$$\max_{\ell \in \{0,\dots,k-1\}} (i_\ell m + i_{\ell+1} n)$$

For S taken as the semiring $\{\mathbb{R}, +, \times, 0, 1\}$, the weight of w would be:

$$\sum_{\ell \in \{0,\dots,k-1\}} m^{i_\ell} n^{i_{\ell+1}}$$

The later is not a probabilistic automaton, as for example there are two transitions labelled by t each of weight 1 exiting q_1.

3 Decision problems

Many decision problems arise naturally when considering weighted automata. We will consider two of them (and their variants): the equivalence and the containment problems. Generally speaking, the decidability of these problems highly depends on the semiring under consideration, and there is no general results that would work for the whole class of weighted automata. However, we will see that some techniques can be used both for max-plus and for probabilistic automata.

3.1 The equivalence and containment problems

The equivalence and containment problems are stated as follows:

> **Equivalence problem**: Given two weighted automata $\mathcal{A}$ and $\mathcal{B}$ over an alphabet Σ and a semiring S, is it true that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$?

This problem has been known to be decidable for Boolean automata since the 50's and for probabilistic automata since the 70's. Surprisingly at the time, it was proved to be undecidable for max-plus automata, in the 90's.

> **Containment problem**: Given two weighted automata $\mathcal{A}$ and $\mathcal{B}$ over an alphabet Σ and a semiring S whose domain is equipped with an order $\leq$, is it true that for all $w \in \Sigma^*$, $[\![\mathcal{A}]\!](w) \leq [\![\mathcal{B}]\!](w)$?

This problem is the counterpart of the containment problem for Boolean automata: given two non deterministic finite automata $\mathcal{A}$ and $\mathcal{B}$, is it true that the regular language accepted by $\mathcal{A}$ is a subset of the regular language accepted by $\mathcal{B}$.

Several other problems of the same kind have been investigated: the boundedness problem for min-plus automata, the isolation, emptiness and value 1 problems for probabilistic automata. They will be discussed later on.

3.2 Undecidability of the containment problem

The containment problem is undecidable both for max-plus and for probabilistic automata and we will explain here one common idea used in specific proofs of this result in both cases.

Probabilistic automata. The more specific emptiness problem:

> *Given a probabilistic automaton $\mathcal{A}$ and a constant c, is it true that for all $w \in \Sigma$, $[\![\mathcal{A}]\!](w) < c$?*

was proved to be undecidable by Paz in 1971 [34]. Other similar problems were also proved to be undecidable like the c-threshold problem [5]:

> *Given a probabilistic automaton $\mathcal{A}$ and a constant c, does there exist $\varepsilon > 0$ such that for all words w, $|[\![\mathcal{A}]\!](w) - c| \geq \varepsilon$?*

and the value 1 problem [19]:

> *Given a probabilistic automaton $\mathcal{A}$ and a constant c, for all $\varepsilon > 0$, does there exist a word w such that $[\![\mathcal{A}]\!](w) \geq 1 - \varepsilon$?*

Max-plus automata. Regarding max-plus automata, the undecidability of the containment problem was first proved in a seminal paper by Krob [28], by reduction from Hilbert's tenth problem: given a diophantine equation, is there a solution with only integral values?

More recently, a new proof has been given by reduction from the halting problem of two-counter machines [1].

More specifically, the two following problems are already undecidable:

- Given a max-plus automaton $\mathcal{A}$ with weights in $\mathbb{Z}$, is it true that for all words w, $[\![\mathcal{A}]\!](w) \geq 0$?
- Given a max-plus automaton $\mathcal{A}$ with weights in $\mathbb{N}$, is it true that for all words w, $[\![\mathcal{A}]\!](w) \geq |w|$?

Restricted classes. Given the negative results stated above, restricted classes of weighted automata have been studied to get decidability. For probabilistic automata one can cite hierarchical [7] and leaktight [16] automata, and both for probabilistic and max-plus automata, classes of automata with restricted ambiguity (see Section 3.4).

However, even for the classes of linearly ambiguous max-plus automata and linearly ambiguous probabilistic automata the containment problem remains undecidable (it is proved in [12] for probabilistic automata, previous proofs were not attaining linear ambiguity - for max-plus automata, this could have already been deduced from the original proof of Krob). In both cases, one can use the reduction from the halting problem of a two counter machine.

Reduction from the halting problem of two counter machines. Two counter machines or Minsky machines have many equivalent definitions. We consider here the following one: A two counter machine is a deterministic finite state machine with two non-negative counters which can be incremented, decremented and checked to be equal to 0. Transitions are of the form $(p, update_c, q)$ where p and q are states of the machine, $i \in \{1, 2\}$ and $update_c$ can be equal to:

- inc_c, meaning that the value of the counter c is incremented,
- dec_c meaning that the value of the counter c is decremented and that the transition can only be taken if the value of the counter c is not 0,
- $check_c$ meaning that the transition can only be taken if the value of the counter c is 0.

Determinism is guaranteed if for every state p, every two distinct transitions exiting from p are labelled by dec_c and $check_c$ respectively for $c = 1$ or $c = 2$. A run of the machine starts from an initial state, follows the unique transitions that can be taken and halts when a final state is reached. If the unique run halts, the machine is said to halt.

The halting problem, i.e. given a two-counter machine, does the unique run halts? is undecidable [31].

Given a two counter machine $\mathcal{M}$, let $\Sigma_{\mathcal{M}}$ be the alphabet containing letters a, b and one letter for each transition of the two-counter machine. The idea

is to encode the value of the first counter into the size of blocks of consecutive occurrences of the letter a and the value of the second counter into the size of blocks of consecutive occurrences of the letter b. More precisely, we say that a word on $\Sigma_{\mathcal{M}}$ encodes a halting run of the machine if it is of the form $a^{i_0}b^{j_0}t_1a^{i_1}b^{j_1}t_2\cdots t_ka^{i_k}b^{j_k}$ where:

1. $t_1, t_2, \ldots, t_k$ are transitions of the machine, such that the sequence forms a valid run with compatible states,
2. $i_0 = 0$, $j_0 = 0$, t_1 starts in the initial state,
3. t_k ends in a final state,
4. for all t_ℓ, if $t_\ell = (p, update_c, q)$, then $(i_{\ell-1}, i_\ell)$ and $(j_{\ell-1}, j_\ell)$ have to be compatible with $update_c$. For example, if $update_c = inc_1$, then we should have $i_\ell = i_{\ell-1} + 1$ and $j_{\ell-1} = j_\ell$. If $update_c = dec_2$ then we should have $i_\ell = i_{\ell-1}$, $j_{\ell-1} \neq 0$ and $j_\ell = j_{\ell-1} - 1$. And similarly for the other cases.

Clearly, a two-counter machine halts if and only if there exist a halting word on $\Sigma_{\mathcal{M}}$. In the case of max-plus automata and probabilistic automata, one can prove that:

Theorem 1 ([1]). *Given a two-counter machine, one can construct a linearly ambiguous max-plus automaton $\mathcal{A}$ such that a word $w \in \Sigma_{\mathcal{M}}$ encodes a halting run if and only if $[\![\mathcal{A}]\!](w) < 0$.*

Theorem 2 ([12]). *Given a two-counter machine, one can construct a linearly ambiguous probabilistic automaton $\mathcal{A}$ such that a word $w \in \Sigma_{\mathcal{M}}$ encodes a halting run if and only if $[\![\mathcal{A}]\!](w) > \frac{1}{2}$.*

This later theorem is also true for all the variants: $\geq$, $<$, $\leq$.

To prove the first result, one needs to construct a max-plus automaton which will give high value (greater than 0) to any word that does not encode a halting run. Or, in other words, a high value to any word that does not satisfy at least one of the conditions 1-4 above. This is done by taking the union of several max-plus automata, each giving high value to words not fulfilling one of the conditions. Conditions 1-3 are regular conditions, and such a construction will be straightforward. Conditions 4 will be dealt with by a similar automaton as the one given in Figure 1.

For probabilistic automata, the idea is similar. However, the construction of an automaton giving high value (greater than $\frac{1}{2}$) to any word encoding a halting run, while ensuring the probabilistic condition is much more intricate than in the max-plus case.

Remark 1. By considering a universal two-counter machine, which encodes initially any two-counter machine in the value of the first counter, these two results prove that the containment problem is also undecidable on classes of max-plus (resp. probabilistic) automata with a bounded number of states.

3.3 Equivalence

For max-plus automata, it is easy to see that the equivalence problem is as difficult as the containment problem, and thus undecidable. This is due to the use of the operation max in the semiring. Indeed, given two automata $\mathcal{A}$ and $\mathcal{B}$, one has that $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ if and only if $[\![\mathcal{B}]\!] = \max([\![\mathcal{A}]\!], [\![\mathcal{B}]\!])$. The later function is the semantics of a max-plus automaton constructible from $\mathcal{A}$ and $\mathcal{B}$ (just taking the union). These problems are also undecidable even when restricting the weights in $\mathbb{N}$.

This is a very different situation for probabilistic automata, as the equivalence is decidable (this is generally true when the domain of the semiring is a field). Tzeng gave an algorithm in PTIME to solve this problem [41].

3.4 Decidability: restricting the ambiguity

On the positive side, containment and equivalence are decidable when restricting sufficiently the ambiguity of the automaton:

Theorem 3 ([24, 43]). *The containment and equivalence problems are decidable on the class of finitely ambiguous max-plus automata.*

For probabilistic automata, the situation is more complex and the decidability of the containment problem on the class of finitely ambiguous probabilistic automata is open. A restricted case is shown to be decidable in [12], and the proof of this result is mathematically difficult.

Theorem 4 ([12]). *If Schanuels conjecture holds then the containment problem is decidable for the class of finitely ambiguous probabilistic automata, provided that at least one of the input automata is unambiguous.*

4 Approximations

Approximations of the containment and equivalence problems have also been studied, in the hope to get decidability, in particular considering algorithms that do not always output the correct result.

4.1 Probabilistic case

Unfortunately, in the probabilistic case, even by allowing quite a lot of flexibility, undecidability persists. Fijalkow gives in [15] a result which subsumes all the results of undecidability for the problems mentioned above (emptiness, value 1 and isolation).

Theorem 5 ([15]). *There exists no algorithm such that: given a probabilistic automaton $\mathcal{A}$,*

- *if $sup_{w\in\Sigma^*}[\![\mathcal{A}]\!](w) = 1$ then the algorithm outputs "Yes",*
- *if $sup_{w\in\Sigma^*}[\![\mathcal{A}]\!](w) \leq \frac{1}{2}$, then the algorithm outputs "No".*

4.2 Approximating max-plus and min-plus automata

For max-plus and min-plus automata, the situation is slightly different and one can obtain good approximations of their semantics provided the weights are restricted to be non-negative.

Since we are now considering only non-negative weights, it is unclear whether the results that are stated below for max-plus automata are also valid for min-plus automata and vice-versa.

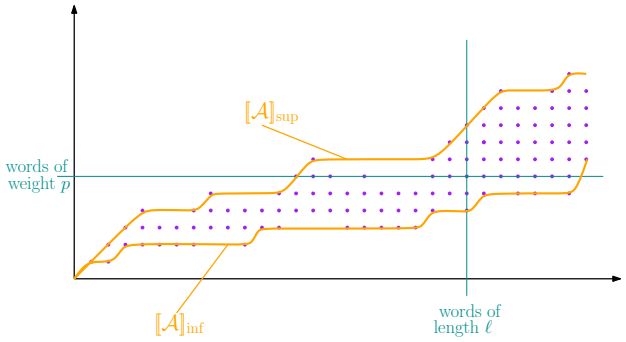
Approximate comparison. The following result shows that an approximation of the containment problem can be decided for min-plus automata. Given two min-plus automata with weights in $\mathbb{N}$, one can compare their semantics up to any small multiplicative error.

Theorem 6 ([8]). *There exist an algorithm which, given $\varepsilon > 0$, given two min-plus automata $\mathcal{A}$ and $\mathcal{B}$ with weights in $\mathbb{N}$, has the following behaviour:*

- *if $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ then the algorithm outputs "Yes",*
- *if there is a word w such that $[\![\mathcal{A}]\!](w) \geq (1+\varepsilon)[\![\mathcal{B}]\!](w)$, then the algorithm outputs "No".*

Note that in the remaining cases, the algorithm can answer "Yes" or "No". It is unclear whether a similar result holds for max-plus automata.

Boundedness and asymptotic descriptions. The semantics of a max-plus or min-plus automaton $\mathcal{A}$ with weights in $\mathbb{N}$ can be represented as follows (let us suppose that no word has weight $-\infty$ or $+\infty$ - it would be easy to reduce the problems to this case).



We are interested in describing the functions $\mathcal{A}_{\sup}$ and $\mathcal{A}_{\inf}$ defined by:

$$\mathcal{A}_{\sup} : \mathbb{N} \to \mathbb{N} \qquad n \mapsto \sup_{w \text{ s.t } |w|=n} [\![\mathcal{A}]\!](w)$$

$$\mathcal{A}_{\inf} : \mathbb{N} \to \mathbb{N} \qquad n \mapsto \inf_{w \text{ s.t } |w|=n} [\![\mathcal{A}]\!](w)$$

The function $\mathcal{A}_{\sup}$ is easy to described for max-plus automata, and similarly the function $\mathcal{A}_{\inf}$ is easy to describe for min-plus automata. However, the other way around is much more difficult.

The boundedness problem asks whether $\mathcal{A}_{\sup}$ is bounded for $\mathcal{A}$ a min-plus automaton. This problem was proved to be decidable by Hashiguchi is 1982 [20, 22, 23].

Theorem 7 ([20]). *The boundedness problem is decidable for min-plus automata with weights in* $\mathbb{N}$.

The function $\mathcal{A}_{\inf}$ can also be described for max-plus automata, and it was proved that this function is asymptotically equivalent to n^α for some rational α in $[0, 1]$. Moreover, all the rationals α in $[0, 1]$ can be attained by some max-plus automata.

Theorem 8 ([9]). *There exist an algorithm which, given a max-plus automaton* $\mathcal{A}$ *with weights in* $\mathbb{N}$*, such that no word is mapped to* $-\infty$*, computes the value* $\alpha \in \{\beta \in \mathbb{Q} \ : \ \beta \in [0, 1]\}$ *such that:* $\mathcal{A}_{\inf} = \Theta(n^\alpha)$.

This result was applied in the automatic analysis of complexity of programs.

All the results given in this section were proved using the forest factorisation theorem and the matrix representation given in Section 2.

5 Conclusion

The containment and equivalence problems have been studied intensively for probabilistic and max-plus automata. Though many open questions remain. First, cost register automata gives an alternative model for weighted automata. Containment and equivalence could be studied on restricted classes of this model. For probabilistic automata, the (un)decidability of the containment problem over the class of finitely ambiguous automata is unknown. For max-plus and min-plus automata, approximations should be generalised and it is unclear if the ones that are decidable for max-plus automata are also decidable for min-plus automata and vice-versa. Finally, generally speaking, complexity issues could be investigated.

References

1. Almagor, S., Boker, U., Kupferman, O.: What's decidable about weighted automata? In: Bultan, T., Hsiung, P. (eds.) Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6996, pp. 482–491. Springer (2011). https://doi.org/10.1007/978-3-642-24372-1_37, https://doi.org/10.1007/978-3-642-24372-1_37

2. Alur, R., D'Antoni, L., Deshmukh, J.V., Raghothaman, M., Yuan, Y.: Regular functions and cost register automata. In: 28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013. pp. 13–22. IEEE Computer Society (2013). https://doi.org/10.1109/LICS.2013.65, https://doi.org/10.1109/LICS.2013.65
3. Alur, R., Raghothaman, M.: Decision problems for additive regular functions. CoRR **abs/1304.7029** (2013), http://arxiv.org/abs/1304.7029
4. Berstel, J., Reutenauer, C.: Recognizable formal power series on trees. Theor. Comput. Sci. **18**, 115–148 (1982). https://doi.org/10.1016/0304-3975(82)90019-6, https://doi.org/10.1016/0304-3975(82)90019-6
5. Bertoni, A., Mauri, G., Torelli, M.: Some recursive unsolvable problems relating to isolated cutpoints in probabilistic automata. In: Salomaa, A., Steinby, M. (eds.) Automata, Languages and Programming, Fourth Colloquium, University of Turku, Finland, July 18-22, 1977, Proceedings. Lecture Notes in Computer Science, vol. 52, pp. 87–94. Springer (1977). https://doi.org/10.1007/3-540-08342-1_7, https://doi.org/10.1007/3-540-08342-1_7
6. Buchsbaum, A.L., Giancarlo, R., Westbrook, J.R.: On the determinization of weighted finite automata. SIAM J. Comput. **30**(5), 1502–1531 (2000). https://doi.org/10.1137/S0097539798346676, https://doi.org/10.1137/S0097539798346676
7. Chadha, R., Sistla, A.P., Viswanathan, M., Ben, Y.: Decidable and expressive classes of probabilistic automata. In: Pitts, A.M. (ed.) Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings. Lecture Notes in Computer Science, vol. 9034, pp. 200–214. Springer (2015). https://doi.org/10.1007/978-3-662-46678-0_13, https://doi.org/10.1007/978-3-662-46678-0_13
8. Colcombet, T., Daviaud, L.: Approximate comparison of distance automata. In: Portier, N., Wilke, T. (eds.) 30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany. LIPIcs, vol. 20, pp. 574–585. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2013). https://doi.org/10.4230/LIPIcs.STACS.2013.574, https://doi.org/10.4230/LIPIcs.STACS.2013.574
9. Colcombet, T., Daviaud, L., Zuleger, F.: Size-change abstraction and max-plus automata. In: Csuhaj-Varjú, E., Dietzfelbinger, M., Ésik, Z. (eds.) Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8634, pp. 208–219. Springer (2014). https://doi.org/10.1007/978-3-662-44522-8_18, https://doi.org/10.1007/978-3-662-44522-8_18
10. Daviaud, L., Guillon, P., Merlet, G.: Comparison of max-plus automata and joint spectral radius of tropical matrices. In: Larsen, K.G., Bodlaender, H.L., Raskin, J. (eds.) 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark. LIPIcs, vol. 83, pp. 19:1–19:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017). https://doi.org/10.4230/LIPIcs.MFCS.2017.19, https://doi.org/10.4230/LIPIcs.MFCS.2017.19
11. Daviaud, L., Jecker, I., Reynier, P., Villevalois, D.: Degree of sequentiality of weighted automata. In: Esparza, J., Murawski, A.S. (eds.) Founda-

tions of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10203, pp. 215–230 (2017). https://doi.org/10.1007/978-3-662-54458-7_13, https://doi.org/10.1007/978-3-662-54458-7_13

12. Daviaud, L., Jurdzinski, M., Lazic, R., Mazowiecki, F., Pérez, G.A., Worrell, J.: When is containment decidable for probabilistic automata? In: Chatzigiannakis, I., Kaklamanis, C., Marx, D., Sannella, D. (eds.) 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic. LIPIcs, vol. 107, pp. 121:1–121:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/LIPIcs.ICALP.2018.121, https://doi.org/10.4230/LIPIcs.ICALP.2018.121
13. Daviaud, L., Reynier, P., Talbot, J.: A generalised twinning property for minimisation of cost register automata. In: Grohe, M., Koskinen, E., Shankar, N. (eds.) Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016. pp. 857–866. ACM (2016). https://doi.org/10.1145/2933575.2934549, https://doi.org/10.1145/2933575.2934549
14. Droste, M., Gastin, P.: Weighted automata and weighted logics. Theor. Comput. Sci. **380**(1-2), 69–86 (2007). https://doi.org/10.1016/j.tcs.2007.02.055, https://doi.org/10.1016/j.tcs.2007.02.055
15. Fijalkow, N.: Undecidability results for probabilistic automata. SIGLOG News **4**(4), 10–17 (2017), https://dl.acm.org/citation.cfm?id=3157833
16. Fijalkow, N., Gimbert, H., Kelmendi, E., Oualhadj, Y.: Deciding the value 1 problem for probabilistic leaktight automata. Logical Methods in Computer Science **11**(2) (2015). https://doi.org/10.2168/LMCS-11(2:12)2015, https://doi.org/10.2168/LMCS-11(2:12)2015
17. Filiot, E., Jecker, I., Lhote, N., Pérez, G.A., Raskin, J.: On delay and regret determinization of max-plus automata. In: 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017. pp. 1–12. IEEE Computer Society (2017). https://doi.org/10.1109/LICS.2017.8005096, https://doi.org/10.1109/LICS.2017.8005096
18. Gaubert, S., Mairesse, J.: Modeling and analysis of timed petri nets using heaps of pieces. IEEE Trans. Automat. Contr. **44**(4), 683–697 (1999). https://doi.org/10.1109/9.754807, https://doi.org/10.1109/9.754807
19. Gimbert, H., Oualhadj, Y.: Probabilistic automata on finite words: Decidable and undecidable problems. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II. Lecture Notes in Computer Science, vol. 6199, pp. 527–538. Springer (2010). https://doi.org/10.1007/978-3-642-14162-1_44, https://doi.org/10.1007/978-3-642-14162-1_44
20. Hashiguchi, K.: Limitedness theorem on finite automata with distance functions. J. Comput. Syst. Sci. **24**(2), 233–244 (1982). https://doi.org/10.1016/0022-0000(82)90051-4, https://doi.org/10.1016/0022-0000(82)90051-4
21. Hashiguchi, K.: Algorithms for determining relative star height and star height. Inf. Comput. **78**(2), 124–169 (1988). https://doi.org/10.1016/0890-5401(88)90033-8, https://doi.org/10.1016/0890-5401(88)90033-8

22. Hashiguchi, K.: Improved limitedness theorems on finite automata with distance functions. Theor. Comput. Sci. **72**(1), 27–38 (1990). https://doi.org/10.1016/0304-3975(90)90044-I, https://doi.org/10.1016/0304-3975(90)90044-I
23. Hashiguchi, K.: New upper bounds to the limitedness of distance automata. In: Meyer auf der Heide, F., Monien, B. (eds.) Automata, Languages and Programming, 23rd International Colloquium, ICALP96, Paderborn, Germany, 8-12 July 1996, Proceedings. Lecture Notes in Computer Science, vol. 1099, pp. 324–335. Springer (1996). https://doi.org/10.1007/3-540-61440-0_139, https://doi.org/10.1007/3-540-61440-0_139
24. Hashiguchi, K., Ishiguro, K., Jimbo, S.: Decidability of the equivalence problem for finitely ambiguous finance automata. IJAC **12**(3), 445 (2002). https://doi.org/10.1142/S0218196702000845, https://doi.org/10.1142/S0218196702000845
25. II, K.C., Kari, J.: Image compression using weighted finite automata. Computers & Graphics **17**(3), 305–313 (1993). https://doi.org/10.1016/0097-8493(93)90079-O, https://doi.org/10.1016/0097-8493(93)90079-O
26. Kirsten, D., Lombardy, S.: Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In: Albers, S., Marion, J. (eds.) 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings. LIPIcs, vol. 3, pp. 589–600. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2009). https://doi.org/10.4230/LIPIcs.STACS.2009.1850, https://doi.org/10.4230/LIPIcs.STACS.2009.1850
27. Klimann, I., Lombardy, S., Mairesse, J., Prieur, C.: Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. Theor. Comput. Sci. **327**(3), 349–373 (2004). https://doi.org/10.1016/j.tcs.2004.02.049, https://doi.org/10.1016/j.tcs.2004.02.049
28. Krob, D.: The equality problem for rational series with multiplicities in the tropical semiring is undecidable. In: Kuich, W. (ed.) Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings. Lecture Notes in Computer Science, vol. 623, pp. 101–112. Springer (1992). https://doi.org/10.1007/3-540-55719-9_67, https://doi.org/10.1007/3-540-55719-9_67
29. Kuich, W.: Formal power series over trees. In: Bozapalidis, S. (ed.) Proceedings of the 3rd International Conference Developments in Language Theory, DLT 1997, Thessaloniki, Greece, July 20-23, 1997. pp. 61–101. Aristotle University of Thessaloniki (1997)
30. Kwiatkowska, M.Z., Norman, G., Parker, D., Qu, H.: Assume-guarantee verification for probabilistic systems. In: Esparza, J., Majumdar, R. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6015, pp. 23–37. Springer (2010). https://doi.org/10.1007/978-3-642-12002-2_3, https://doi.org/10.1007/978-3-642-12002-2_3
31. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Inc., Englewood Cliffs, N.J. (1967), prentice-Hall Series in Automatic Computation
32. Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics **23**(2), 269–311 (1997)

33. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. Computer Speech & Language **16**(1), 69–88 (2002). https://doi.org/10.1006/csla.2001.0184, https://doi.org/10.1006/csla.2001.0184
34. Paz, A.: Introduction to Probabilistic Automata (Computer Science and Applied Mathematics). Academic Press, Inc. (1971)
35. Rabin, M.O.: Probabilistic automata. Information and Control **6**(3), 230–245 (1963). https://doi.org/10.1016/S0019-9958(63)90290-0, https://doi.org/10.1016/S0019-9958(63)90290-0
36. Sakarovitch, J.: Elements of Automata Theory. Cambridge University Press (2009), http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521844253
37. Schützenberger, M.P.: On the definition of a family of automata. Information and Control **4**(2-3), 245–270 (1961). https://doi.org/10.1016/S0019-9958(61)80020-X, https://doi.org/10.1016/S0019-9958(61)80020-X
38. Seidl, H.: Finite tree automata with cost functions. Theor. Comput. Sci. **126**(1), 113–142 (1994). https://doi.org/10.1016/0304-3975(94)90271-2, https://doi.org/10.1016/0304-3975(94)90271-2
39. Simon, I.: Limited subsets of a free monoid. In: 19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978. pp. 143–150. IEEE Computer Society (1978). https://doi.org/10.1109/SFCS.1978.21, https://doi.org/10.1109/SFCS.1978.21
40. Simon, I.: Recognizable sets with multiplicities in the tropical semiring. In: Chytil, M., Janiga, L., Koubek, V. (eds.) Mathematical Foundations of Computer Science 1988, MFCS'88, Carlsbad, Czechoslovakia, August 29 - September 2, 1988, Proceedings. Lecture Notes in Computer Science, vol. 324, pp. 107–120. Springer (1988). https://doi.org/10.1007/BFb0017135, https://doi.org/10.1007/BFb0017135
41. Tzeng, W.: A polynomial-time algorithm for the equivalence of probabilistic automata. SIAM J. Comput. **21**(2), 216–227 (1992). https://doi.org/10.1137/0221017, https://doi.org/10.1137/0221017
42. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. In: 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985. pp. 327–338. IEEE Computer Society (1985). https://doi.org/10.1109/SFCS.1985.12, https://doi.org/10.1109/SFCS.1985.12
43. Weber, A.: Finite-valued distance automata. Theor. Comput. Sci. **134**(1), 225–251 (1994). https://doi.org/10.1016/0304-3975(94)90287-9, https://doi.org/10.1016/0304-3975(94)90287-9