Filomat 40:10 (2026), 3499–3521
https://doi.org/10.2298/FIL2610499M

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Weak simulations and bisimulations for max-plus automata

Jelena Matejić[a,*], Ivana Micić[a], Miroslav Ćirić[a]

[a] *University of Niš, Faculty of Sciences and Mathematics, Department of Computer Science, Višegradska 33, 18000 Niš, Serbia*

Abstract. In this paper, for the first time in the context of max-plus automata, we introduce two types of weak simulations and two types of weak bisimulations. Each of these is a generalization of the corresponding type of standard simulation or bisimulation that were previously introduced and studied in [9]. We show that weak simulations of any type can be used to confirm containment between max-plus automata, while weak bisimulations of any type can be used to confirm equivalence. Moreover, they apply in a strictly broader range of situations than the corresponding standard simulations and bisimulations, making them a more powerful tool for reasoning about quantitative behavioral relations. We also provide a procedure for testing the existence of a weak simulation or bisimulation of a given type, which – whenever the existence is confirmed – simultaneously computes the greatest weak simulation or bisimulation of that type.

1. Introduction

Max-plus automata are weighted finite automata whose computations take place in the max-plus semiring, where the role of addition is played by the maximum operation, while multiplication is realized as ordinary addition. This algebraic setting has proven highly effective for modeling and analyzing systems in which timing, synchronization, and resource sharing play a crucial role. Typical application domains include manufacturing systems, transport and traffic systems, communication protocols, scheduling tasks, and more broadly, discrete-event systems. In such domains, max-plus operations naturally reflect how delays accumulate, how processing times propagate through a system, and how parallel events synchronize, making max-plus models both conceptually clear and mathematically manageable. Max-plus automata can be interpreted as quantitative models of timed behavior, where transition weights represent durations, costs, or delays, and the value assigned to a word corresponds to the maximal accumulated weight among all runs labeled by that word. Under this interpretation, max-plus automata characterize the worst-case or longest execution time associated with a particular sequence of events. This makes them particularly useful in performance evaluation and verification, where extreme timing scenarios are often the main focus. Thanks to their robust algebraic foundations and expressive capabilities, max-plus automata occupy

2020 *Mathematics Subject Classification*. Primary 68Q70; Secondary 15A80, 16Y60, 93C65.
Keywords. complete max-plus semiring, weighted automata, max-plus automata, weak simulations, weak bisimulations
Received: 30 August 2025; Revised: 22 December 2025; Accepted: 11 February 2026
Communicated by Dragan S. Djordjević
Support was provided by the Ministry of Science, Technological Development, and Innovation of the Republic of Serbia under the 2026 Research Program (Agreement No. 451-03-34/2026-03/200124).
* Corresponding author: Jelena Matejić
Email addresses: jelena.matejic@pmf.edu.rs (Jelena Matejić), ivana.micic@pmf.edu.rs (Ivana Micić), miroslav.ciric@pmf.edu.rs (Miroslav Ćirić)
ORCID iDs: https://orcid.org/0000-0001-6410-4296 (Jelena Matejić), https://orcid.org/0000-0002-3816-9464 (Ivana Micić), https://orcid.org/0000-0001-8625-4682 (Miroslav Ćirić)

an important place at the intersection of automata theory, algebra, and systems theory. Understanding their properties, equivalences, and computational aspects is therefore of both theoretical significance and practical relevance. For more detailed information on max-plus automata, as well as bibliographic notes about them, we refer to the papers [3, 9, 10, 12–14, 17, 25–32, 43–45].

Two central *decision problems* for weighted finite automata are the equivalence problem and the containment problem. For a pair of weighted automata defined over the same semiring, the *equivalence problem* asks whether both automata assign exactly the same quantitative value to every input word. The *containment problem*, on the other hand, assumes that the underlying semiring is equipped with a compatible order and asks whether, for every word, the value produced by one automaton is always less than or equal to the value assigned by the other. These notions extend the classical problems of language equivalence and inclusion to the quantitative setting, and they serve as fundamental tools for verification, optimization, and comparative analysis of models. When specialized to max-plus automata, these decision problems admit a particularly intuitive interpretation. Since weights represent durations, costs, or delays, containment asks whether one system is guaranteed to be no slower (in the worst case) than another, while equivalence captures the requirement that both systems exhibit exactly the same worst-case timing behavior for every possible sequence of events. These interpretations make the equivalence and containment problems especially relevant for performance analysis and refinement checking in timed and discrete-event systems.

Despite their conceptual simplicity, both problems are computationally challenging. In the max-plus setting, equivalence is known to be decidable under certain restrictions, but it becomes difficult or undecidable in more general frameworks (cf. [13, 14]). Similar difficulties arise for containment, which is often harder than equivalence due to its inherently asymmetric nature. To address these challenges, *simulation* and *bisimulation relations* have been introduced as sound proof techniques for establishing containment and equivalence. Simulations were introduced to formalize *behavioral containment*: the idea that every observable action of one system can be matched by a corresponding action of another system. Bisimulations, on the other hand, were developed to formalize *behavioral equivalence*, ensuring that two systems can mutually mimic each other's behavior step by step. These notions emerged in the context of concurrency theory and process algebra, where they offered a rigorous foundation for reasoning about system correctness, abstraction, minimization, and compositional verification. Over time, simulations and bisimulations have become central concepts in automata theory, model checking, and formal verification, precisely because they provide constructive certificates of containment and equivalence between computational models.

When quantitative systems, such as weighted automata, probabilistic systems, or timed systems, came into the focus of research instead of classical qualitative systems, it became clear that classical relation-based simulations and bisimulations were not capable of handling the quantitative properties of these systems. Consequently, there emerged a need to generalize classical notions of simulation and bisimulation to handle quantitative aspects effectively. Quantitative simulations and bisimulations extend the classical concepts by integrating algebraic or numerical structures into the relational framework. They allow the comparison of systems not only in terms of their qualitative behavior but also in terms of numerical measures, such as accumulated weights or probabilities along executions. These notions have become crucial tools in the analysis and verification of quantitative systems, providing formal means to check behavioral containment, equivalence, and performance properties. In the context of max-plus automata, a quantitative approach was developed in [9, 31, 44], where simulations and bisimulations are defined as matrices over the max-plus semiring rather than as Boolean relations or Boolean matrices. A general theory of simulations and bisimulations for max-plus automata was developed in [9], where two types of simulations and four types of bisimulations were defined as solutions of specific systems of matrix inequalities, thereby reducing the problem of testing the existence of a simulation or bisimulation of a given type, as well as computing the greatest simulation or bisimulation of that type, if it exists, to the problem of solving the corresponding system of matrix inequalities. In the same paper, algorithms for solving the aforementioned problems were also developed, and some improvements of these algorithms were proposed recently in [33].

It should be noted that simulations for max-plus automata first appeared in [44], where they were defined as matrices over the max-plus semiring that are solutions of systems of matrix inequalities similar (though not identical) to those presented in [9]. That paper established a connection between simulations and mean payoff games and proposed an algorithm for testing the existence and computing simulations, which reduces this problem to the solution of two-sided linear equation over the max-plus semiring (cf. [2, 5]). Defined in the same way, but as Boolean matrices (i.e., as ordinary binary relations), simulations also appeared in [26], where they were used in the study of max-plus automata with partial observations, and in [31], where they were employed in the determinization of max-plus automata. It should be noted that these papers relied on the results from [11] concerning simulations considered in the somewhat more general context of weighted finite automata over an additively idempotent semiring, which are also defined as Boolean matrices. In the second part of [31], simulations were also considered as matrices over the max-plus semiring, and it was shown that the problem of the existence of simulations can be reduced to the problem of the non-emptiness of a tropical polyhedron (the set of solutions of a two-sided linear inequality over the max-plus semiring). The conditions for the existence of a simulation between max-plus automata were also studied in [12]. As we have already noted, the iterative approach to solving the systems of matrix inequalities that define simulations and bisimulations, which we use here, was introduced in [9] and is based on the approach to solving somewhat more general weakly linear systems of matrix inequalities developed in [38].

In this paper, we introduce the notions of *weak simulations* and *weak bisimulations*, which are new in the context of max-plus automata, although they have already been studied in the setting of fuzzy automata. Within the theory of fuzzy automata, these notions were first introduced in [21], and apart from being used for confirming containment and equivalence, they also proved to be highly effective in state reduction and determinization (cf. [21–24, 34, 35, 39, 40]). We introduce two types of weak simulations – *forward* and *backward*, as well as two types of weak bisimulations – also *forward* and *backward*, and by slightly relaxing their definitions, we define the corresponding types of *weak presimulations* and *weak prebisimulations*. What we first prove is that weak presimulations of a given type form a generalization of the standard presimulations of the same type, in the sense that every presimulation of that type is also a weak presimulation of that type. The same has been proved for weak simulations, as well as for weak prebisimulations and weak bisimulations (Theorem 4.1). After that, we prove that the existence of a weak simulation of any type between two max-plus automata provides confirmation of the containment relation between these automata, whereas the existence of a weak bisimulation of any type provides confirmation of their equivalence (Theorem 4.2). This result underpins the primary role of weak simulations and weak bisimulations.

Just like their standard counterparts, weak simulations and weak bisimulations of any type are also defined by means of particular systems of matrix inequalities. The inequalities that constitute any of these systems can be divided into two groups. The system consisting of the inequalities from the second group, which defines a weak presimulation or weak prebisimulation of the given type, is always solvable and has the greatest solution – the greatest weak presimulation or weak prebisimulation of that type. If this greatest solution is also a solution of the system consisting of the inequalities from the first group, then it is the greatest weak simulation or greatest weak bisimulation of that type, and otherwise, if it is not a solution of that system, then no weak simulation or weak bisimulation of that type exists. This is established for each type of weak simulation and weak bisimulation by Theorem 4.5. The same theorem also provides a procedure for computing the greatest weak presimulation or weak prebisimulation of an arbitrary type, which is presented in detail in Section 5 as Procedure 1. The main difficulty in applying this procedure is that it may fail to terminate in a finite number of steps. This occurs when the system being solved contains infinitely many inequalities. Examples 5.1 and 5.4 illustrate situations in which the procedure for computing the greatest forward presimulation terminates after finitely many steps, whereas Examples 5.2 and 5.3 demonstrate cases in which the system consists of infinitely many inequalities and cannot be solved by Procedure 1 in a finite number of steps. Nevertheless, in these cases solutions were still obtained by applying mathematical induction. In any case, the problem of a potentially infinite number of inequalities to be solved deserves special attention and will be addressed in our future research.

The computational examples we provide highlight yet another potential issue that warrants attention. The matrix computed by the procedure is the infimum of a family of matrices. If this family is finite, the infimum can be obtained in finitely many steps: we start from an initial matrix, and at each subsequent step the current matrix is updated, so that upon termination we obtain the desired infimum. However, Example 5.1 shows that the infimum may in fact be reached already in the very first steps, making all further computations redundant. In our future research, we will attempt to identify criteria for detecting when the infimum has already been reached, in order to avoid unnecessary computations.

A drawback of weak simulations and bisimulations is that, in general, they are more difficult to compute than their standard counterparts, due, as we have already mentioned, to the potentially infinite number of inequalities that need to be solved. However, weak simulations and bisimulations also offer advantages. The main advantage is that they cover a broader range of cases in which they can be applied to confirm containment and equivalence compared to standard simulations and bisimulations. This is because there exist situations where a weak simulation or bisimulation of a given type exists, while a standard simulation or bisimulation of that type does not, as illustrated in Example 5.4, and the converse is not possible, in accordance with Theorem 4.1. In our future research, we will also show that, when used for state-space reduction or determinization of max-plus automata, weak simulations and bisimulations yield automata of smaller size than those obtained using standard simulations and bisimulations.

Finally, as Example 5.1 demonstrates, situations may occur in which one type of weak simulation or bisimulation exists while another does not. This shows that the different types are independent, and that no single type can be given priority a priori in applications. Rather, if one type fails to exist, we simply attempt to find another.

The paper consists of five sections. After this introductory section, Section 2 introduces the basic concepts and notation related to semirings, complete additively idempotent semirings and the complete max-plus semiring, as well as the concepts and notation concerning vectors and matrices over the complete max-plus semiring. In Section 3, we recall the basic concepts and notation related to weighted automata over a semiring and max-plus automata. The concepts of weak simulations and weak bisimulations are introduced in Section 4, where the aforementioned main results of the paper are also presented. Finally, Section 5 describes the procedure for computing the greatest weak presimulations and pre-bisimulations, as well as the procedure for testing the existence of weak simulations and bisimulations, and provides computational examples illustrating the application of these procedures.

Finally, it should be noted that the results of this paper are formulated and proved for weighted automata over the complete max-plus semiring, but they can be established without significant changes for weighted automata over any commutative complete additively idempotent semiring. Moreover, with minor modifications, the same results can also be obtained for weighted automata over any complete additively idempotent semiring (not necessarily commutative).

2. Preliminaries

Throughout this paper, $\mathbb{N}$ denotes the set of natural numbers, $\mathbb{N}_0$ the set of natural numbers including zero, $\mathbb{Z}$ the set of integers, and $\mathbb{R}$ the set of real numbers. We also set $\mathbb{R}_\infty = \mathbb{R} \cup \{-\infty, +\infty\}$, $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$, $\mathbb{Z}_\infty = \mathbb{Z} \cup \{-\infty, +\infty\}$ and $\mathbb{Z}_{\max} = \mathbb{Z} \cup \{-\infty\}$. The standard linear order $\leqslant$ of the set of real numbers is extended to $\mathbb{R}_\infty$ and $\mathbb{R}_{\max}$ so that $-\infty$ is the least element in both $\mathbb{R}_\infty$ and $\mathbb{R}_{\max}$, and $+\infty$ is the greatest element in $\mathbb{R}_\infty$. Without risk of confusion, this extended linear order is also denoted by the same symbol $\leqslant$. For $n \in \mathbb{N}$, by $[1..n]$ we denote the set of all natural numbers between 1 and n.

2.1. *Complete additively idempotent semirings. The complete max-plus semiring*

A *semigroup* is an algebraic structure $(\mathbb{S}, \otimes)$, with a non-empty carrier set $\mathbb{S}$ and an associative binary operation $\otimes$, called the *multiplication*. A *monoid* is an algebraic structure $(\mathbb{S}, \otimes, e)$ with a binary operation $\otimes$ and a constant $e \in \mathbb{S}$ such that $(\mathbb{S}, \otimes)$ is a semigroup and $e \otimes x = x \otimes e = x$, for every $x \in \mathbb{S}$. The element e is called the *identity* of that monoid, or the *neutral element* for the operation $\otimes$. We mainly use the latter term in situations where we work with multiple operations that have different neutral elements. If $\otimes$ is a commutative operation, then $(\mathbb{S}, \otimes)$ is called a *commutative semigroup* and $(\mathbb{S}, \otimes, e)$ a *commutative monoid*.

A *semiring* is an algebraic structure $\mathbb{S} = (\mathbb{S}, \oplus, \otimes, 0, 1)$ with a carrier set $\mathbb{S}$, two binary operations $\oplus$ and $\otimes$ on $\mathbb{S}$, and two constants $0, 1 \in \mathbb{S}$ which satisfy the following conditions:

(S1) $(\mathbb{S}, \oplus, 0)$ is a commutative monoid with neutral element 0, (S2) $(\mathbb{S}, \otimes, 1)$ is a monoid with neutral element 1,

(S3) The *distributivity laws* $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ and $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$ hold for all $a, b, c \in \mathbb{S}$,

(S4) The *absorbtion law* $0 \otimes a = a \otimes 0 = 0$ holds for all $a \in \mathbb{S}$.

The operation $\oplus$ is called the *addition*, and the operation $\otimes$ the *multiplication* of the semiring $\mathbb{S}$. The structure $(\mathbb{S}, \oplus, 0)$ is called the *additive monoid*, while $(\mathbb{S}, \otimes, 1)$ is called the *multiplicative monoid* of the semiring $\mathbb{S}$. The neutral element for multiplication is called the *identity* of the semiring $\mathbb{S}$, while the neutral element for addition is called the *zero* of the semiring $\mathbb{S}$. According to (S4), the zero 0 is the *absorbing element* for multiplication. A semiring is *commutative* if its multiplicative monoid is commutative. It is customary to identify an algebraic structure with its carrier set, so a semiring and its carrier set are denoted by the same symbol. By an *ordered semiring* we mean a semiring equipped with a partial order $\leqslant$ that is compatible with addition and with multiplication by elements greater than zero.

A semiring $\mathbb{S}$ is called *additively idempotent* if its addition is an idempotent operation, that is, $a \oplus a = a$, for all $a \in \mathbb{S}$. On an arbitrary additively idempotent semiring $\mathbb{S}$, one can define the *canonical partial order* $\leqslant$ by

$$a \leqslant b \quad \Leftrightarrow \quad a \oplus b = b, \qquad \text{for all } a, b \in \mathbb{S}.$$

The canonical partial order is compatible both with addition and multiplication, and 0 is the least element. In addition, $(\mathbb{S}, \oplus)$ forms an *upper semilattice*, which means that every two-element subset $\{a, b\}$ of $\mathbb{S}$ has a supremum, which is precisely the sum $a \oplus b$ (cf. [20]). This, in turn, implies that every finite subset $\{a_1, a_2, \ldots, a_k\}$ of $\mathbb{S}$ also has a supremum, which is equal to $a_1 \oplus a_2 \oplus \cdots \oplus a_k$. However, infinite subsets of an additively idempotent semiring $\mathbb{S}$ do not necessarily have suprema in the general case. If $\mathbb{S}$ is an additively idempotent semiring in which every subset $\{a_i\}_{i \in I} \subseteq \mathbb{S}$ has a supremum, this supremum will be denoted by $\bigoplus_{i \in I} a_i$ and will be treated as an infinite sum (it satisfies all conditions from the definition of infinite sums, cf. [15]). In this case we say that $(S, \oplus)$ is a *complete upper semilattice*. Moreover, since this complete upper semilattice has the least element 0, it is also a *complete lattice*, i.e., every subset $\{a_i\}_{i \in I} \subseteq \mathbb{S}$ has an infimum, which we denote by $\bigwedge_{i \in I} a_i$. If $\mathbb{S}$ is an additively idempotent semiring such that $(\mathbb{S}, \oplus)$ is a complete upper semilattice and the *infinite distributive laws*

$$a \otimes \Big(\bigoplus_{i \in I} b_i \Big) = \bigoplus_{i \in I} (a \otimes b_i), \quad \Big(\bigoplus_{i \in I} b_i \Big) \otimes a = \bigoplus_{i \in I} (b_i \otimes a), \tag{1}$$

hold for all $a \in \mathbb{S}$ and $\{b_i\}_{i \in I} \subseteq \mathbb{S}$, then we call $\mathbb{S}$ a *complete additively idempotent semiring*. From the viewpoint of lattice theory, such an algebraic structure is called a *unital quantale* (cf. [16, 36, 37]).

One of the most important properties of a complete additively idempotent semiring $\mathbb{S}$ is the existence of a pair of binary operations $\backslash$ and $/$ on $\mathbb{S}$ such that, for arbitrary $a, b \in \mathbb{S}$, the elements $a \backslash b$ and b / a represent the greatest solutions of the inequalities $a \otimes x \leqslant b$ and $x \otimes a \leqslant b$, respectively, where x denotes the unknown taking values in $\mathbb{S}$. We call $a \backslash b$ the *left residual* of b by a, and b / a the *right residual* of b by a, and the operations $\backslash$ and $/$ are called *residuum operations* adjoined to the multiplication $\otimes$. The relationship among these operations is determined by the *residuation property* (also known as the *adjunction property*), which states that

$$a \otimes b \leqslant c \quad \Leftrightarrow \quad b \leqslant a \backslash c \quad \Leftrightarrow \quad a \leqslant c / b, \tag{2}$$

for all $a, b, c \in \mathbb{S}$. In the case when the complete additively idempotent semiring $\mathbb{S}$ is commutative, the residuum operations $\backslash$ and $/$ coincide and form a single binary operation, denoted by $\to$. This means that $a \to b = a \backslash b = b / a$, for all $a, b \in \mathbb{S}$. The operation $\to$ is called the *residuum operation adjoined to* the multiplication $\otimes$.

This paper focuses on the semiring $\mathbb{R}_\infty = (\mathbb{R}_\infty, \oplus, \otimes, -\infty, 0)$, where the addition operation $\oplus$ is defined by $a \oplus b = \max(a, b)$, for all $a, b \in \mathbb{R}_\infty$, and the multiplication operation $\otimes$ is defined by the following Cayley-like table

$\otimes$	$b \in \mathbb{R}$	$-\infty$	$+\infty$
$a \in \mathbb{R}$	$a + b$	$-\infty$	$+\infty$
$-\infty$	$-\infty$	$-\infty$	$-\infty$
$+\infty$	$+\infty$	$-\infty$	$+\infty$

Clearly, $-\infty$ is the zero and 0 is the identity of this semiring. We call $\mathbb{R}_\infty$ the *complete max-plus semiring*, while its subsemiring $\mathbb{R}_{\max}$ is referred to simply as the *max-plus semiring*. The semiring $\mathbb{R}_{\max}$ is also known as the *max-plus algebra* or the *max-algebra*. Max-plus semirings belong to the most powerful mathematical tools for modeling and analyzing systems in which resource sharing, timing, and synchronization are fundamental. Typical application domains include production systems, transportation networks, communication protocols, scheduling problems, discrete-event systems, and others. The complete max-plus semiring $\mathbb{R}_\infty$ is a commutative complete additively idempotent semiring (which is the origin of the prefix *complete* in its name), whereas the max-plus semiring $\mathbb{R}_{\max}$ is a commutative additively idempotent semiring, but it is not complete. It is important to note that the canonical partial order on the additively idempotent semirings $\mathbb{R}_\infty$ and $\mathbb{R}_{\max}$ coincides with the usual order on the real numbers, extended to $\mathbb{R}_\infty$ and $\mathbb{R}_{\max}$ in the manner described at the beginning of this section.

Considering that $\mathbb{R}_\infty$ is a commutative complete additively idempotent semiring, it possesses a single residuum operation $\to$ adjoined to the multiplication, given by the following Cayley-like table

$\to$	$b \in \mathbb{R}$	$-\infty$	$+\infty$
$a \in \mathbb{R}$	$b - a$	$-\infty$	$+\infty$
$-\infty$	$+\infty$	$+\infty$	$+\infty$
$+\infty$	$-\infty$	$-\infty$	$+\infty$

In the examples provided at the end of this paper, we also consider the semiring $\mathbb{Z}_\infty = (\mathbb{Z}_\infty, \oplus, \otimes, -\infty, 0)$, which is also a commutative complete additively idempotent semiring. Since $\mathbb{Z}_\infty$ is closed under all basic operations of $\mathbb{R}_\infty$, as well as under infinite sums, infima, and the residuum operation in $\mathbb{R}_\infty$, all basic and derived operations in $\mathbb{Z}_\infty$ are denoted by the same symbols as the corresponding operations in $\mathbb{R}_\infty$. We call $\mathbb{Z}_\infty$ the *complete max-plus semiring of integers*.

For more detailed information on the max-plus semiring and its applications, we refer the reader to [1, 4, 19].

2.2. *Matrices and vectors with entries in the complete max-plus semiring*

Throughout the paper, for $m, n \in \mathbb{N}$, the set of all $m \times n$ matrices with entries in $\mathbb{R}_\infty$ is denoted by $\mathbb{R}_\infty^{m\times n}$, and the set of all vectors of size n with entries in $\mathbb{R}_\infty$ is denoted by $\mathbb{R}_\infty^n$. When vectors from $\mathbb{R}_\infty^n$ are written as *row vectors* ($1 \times n$ matrices), then instead of $\mathbb{R}_\infty^n$ we write $\mathbb{R}_\infty^{1\times n}$, and when they are written as *column vectors* ($n \times 1$ matrices), then instead of $\mathbb{R}_\infty^n$ we write $\mathbb{R}_\infty^{n\times 1}$. For a matrix $M \in \mathbb{R}_\infty^{m\times n}$, the entry of M located in the i-th row and j-th column is denoted by $M(i, j)$, and for a vector $\alpha \in \mathbb{R}_\infty^n$, the i-th entry of α is denoted by $\alpha(i)$.

An $m \times n$ matrix whose all entries are equal to $-\infty$, denoted by $O_{m\times n}$, is called the *zero matrix* of type $m \times n$, and a vector of size n whose all entries are equal to $-\infty$ is called the *zero vector* of that size. On the other hand, for any $n \in \mathbb{N}$, the matrix $I_n \in \mathbb{R}_\infty^{n\times n}$ defined by

$$I_n(i, j) = \begin{cases} 0 & \text{if } i = j \\ -\infty & \text{if } i \neq j \end{cases},$$

for all $i, j \in [1..n]$, is called the *identity matrix* of order n. The *transpose* of a matrix $M \in \mathbb{R}_\infty^{m\times n}$ is a matrix $M^\top \in \mathbb{R}_\infty^{n\times m}$ obtained by interchanging the rows and the columns of M, i.e., $M^\top(j, i) = M(i, j)$, for all $i \in [1..m]$ and $j \in [1..n]$.

For an arbitrary family (either finite or infinite) of $m \times n$ matrices $\{M_s\}_{s\in I}$, their *sum* is the $m \times n$ matrix $\bigoplus_{s\in I} M_s$ defined pointwise by

$$\Big(\bigoplus_{s\in I} M_s\Big)(i, j) = \bigoplus_{s\in I} M_s(i, j), \tag{3}$$

for all $i \in [1..m]$ and $j \in [1..n]$. When the index set I is finite we obtain a finite sum.

For matrices $M \in \mathbb{R}_\infty^{m\times n}$ and $N \in \mathbb{R}_\infty^{n\times p}$, their *matrix product* is a matrix $M \otimes N \in \mathbb{R}_\infty^{m\times p}$ defined by

$$(M \otimes N)(i,k) = \bigoplus_{j=1}^{n} \Big(M(i,j) \otimes M(j,k) \Big), \tag{4}$$

for all $i \in [1..m]$ and $k \in [1..p]$. Futher, for a row vector $\alpha \in \mathbb{R}_\infty^{1\times m}$, a column vector $\beta \in \mathbb{R}_\infty^{n\times 1}$, and a matrix $M \in \mathbb{R}_\infty^{m\times n}$, the *vector-matrix product* $\alpha \otimes M \in \mathbb{R}_\infty^{1\times n}$ and the *matrix-vector product* $M \otimes \beta \in \mathbb{R}_\infty^{m\times 1}$ are vectors defined by

$$(\alpha \otimes M)(j) = \bigoplus_{k=1}^{m} \Big(\alpha(k) \otimes M(k,j) \Big), \qquad (M \otimes \beta)(i) = \bigoplus_{k=1}^{n} \Big(M(i,k) \otimes \beta(k) \Big), \tag{5}$$

for all $i \in [1..m]$ and $j \in [1..n]$. Finally, the *scalar product* (or *dot product*) $\alpha \otimes \beta$ of a row vector $\alpha \in R_\infty^{1\times n}$ and a column vector $\beta \in R_\infty^{n\times 1}$ is an element of R_∞ given by

$$\alpha \otimes \beta = \bigoplus_{i=1}^{n} \Big(\alpha(i) \otimes \beta(i) \Big). \tag{6}$$

For matrices of the same type $M, N \in R_\infty^{m\times n}$, the *matrix ordering* is defined coordinatewise, as follows

$$M \leqslant N \quad \Leftrightarrow \quad M(i,j) \leqslant N(i,j), \ \text{ for all } i \in [1..m] \text{ and } j \in [1..n]. \tag{7}$$

In the same way, for vectors of the same size, the coordinatewise *ordering of vectors* is defined. With respect to this matrix ordering, $R_\infty^{m\times n}$ forms a complete lattice in which the supremum of an any family of matrices $\{M_s\}_{s\in I} \subseteq R_\infty^{m\times n}$ is the sum $\bigoplus_{s\in I} M_s$ of that family, while the infimum of that family is a matrix $\bigwedge_{s\in I} M_s \in R_\infty^{m\times n}$ that is also defined pointwise, as follows:

$$\Big(\bigwedge_{s\in I} M_s \Big)(i,j) = \bigwedge_{s\in I} M_s(i,j), \tag{8}$$

for all $i \in [1..m]$ and $j \in [1..n]$. For convenience, when matrices $M, N \in \mathbb{R}_\infty^{m\times n}$ satisfy $M \leqslant N$, we say that *M is contained in N*.

For matrices $M \in R_\infty^{m\times n}$ and $P \in R_\infty^{m\times p}$, the greatest solution of the matrix inequality $M \otimes U \leqslant P$, where U is an unknown matrix taking values in $R_\infty^{n\times p}$, is the matrix $M \backslash P \in R_\infty^{n\times p}$ defined by

$$(M \backslash P)(j,k) = \bigwedge_{i=1}^{m} M(i,j) \to P(i,k), \tag{9}$$

for all $j \in [1..n]$ and $k \in [1..p]$. We call $M \backslash P$ the *left residual* od P by M. Similarly, for matrices $N \in R_\infty^{n\times p}$ and $P \in R_\infty^{m\times p}$, the greatest solution of the matrix inequality $U \otimes N \leqslant P$, with an unknown matrix U taking values in $R_\infty^{m\times n}$, is the matrix $P / N \in R_\infty^{m\times n}$ defined by

$$(P / N)(i,j) = \bigwedge_{k=1}^{p} N(j,k) \to P(i,k), \tag{10}$$

for all $i \in [1..m]$ and $j \in [1..n]$. The matrix P / N is called the *right residual* of P by N. The matrix residuum operations defined in this way satisfy the following *residuation property* (*adjunction property*):

$$M \otimes N \leqslant P \quad \Leftrightarrow \quad N \leqslant M \backslash P \quad \Leftrightarrow \quad M \leqslant P / N, \tag{11}$$

for all $M \in \mathbb{R}_\infty^{m\times n}$, $N \in \mathbb{R}_\infty^{n\times p}$ and $P \in \mathbb{R}_\infty^{m\times p}$.

On the other hand, for vectors $\alpha \in \mathbb{R}_\infty^m$ and $\beta \in \mathbb{R}_\infty^n$, considered as row vectors, the greatest solution of the vector-matrix inequality $\alpha \otimes U \leqslant \beta$, where U is an unknown matrix taking values in $\mathbb{R}_\infty^{m\times n}$, is the matrix $\alpha \backslash \beta \in \mathbb{R}_\infty^{m\times n}$ defined by

$$(\alpha \backslash \beta)(i,j) = \alpha(i) \backslash \beta(j), \tag{12}$$

for all $i \in [1..m]$ and $j \in [1..n]$. We call $\alpha \backslash \beta$ the *left residual* of β by α. Similarly, if the same vectors are considered as column vectors, the greatest solution of the vector-matrix inequality $V \otimes \alpha \leqslant \beta$, where V is an unknown matrix taking values in $\mathbb{R}_\infty^{n\times m}$, is the matrix $\beta / \alpha \in \mathbb{R}_\infty^{n\times m}$ defined by

$$(\beta / \alpha)(j,i) = \beta(j) / \alpha(i), \tag{13}$$

for all $j \in [1..n]$ and $i \in [1..m]$. The matrix $\alpha \backslash \beta$ is called the *right residual* of β by α. Due to the commutativity of the multiplication $\otimes$, one can easily show that $\beta / \alpha = (\alpha \backslash \beta)^\top$, that is, $\alpha \backslash \beta = (\beta / \alpha)^\top$.

The *residuation property* for this kind of residuum operations is

$$\alpha \otimes U \leqslant \beta \quad \Leftrightarrow \quad U \leqslant \alpha \backslash \beta, \qquad V \otimes \alpha \leqslant \beta \quad \Leftrightarrow \quad V \leqslant \beta / \alpha, \tag{14}$$

for all $\alpha \in \mathbb{R}_\infty^m$, $\beta \in \mathbb{R}_\infty^n$, $U \in \mathbb{R}_\infty^{m\times n}$ and $V \in \mathbb{R}_\infty^{n\times m}$.

In addition to the left and right residuals, for vectors $\alpha \in \mathbb{R}_\infty^m$ and $\beta \in \mathbb{R}_\infty^n$, we also define the *biresidual* $\alpha \mid \beta$ of α and β by setting $\alpha \mid \beta = \alpha \backslash \beta \wedge \alpha / \beta$.

3. Max-plus automata

Let X a non-empty set, which we call an *alphabet* and whose elements we call *letters*. The set of all finite sequences of letters from X written in the form $x_1x_2\dots x_k$, including the empty sequence, is denoted by X^*, whereas the set of all non-empty sequences from X^* is denoted by X^+. The elements of X^* are called *words* over X. Equipped with the *concatenation operation*, the set X^* is a monoid, called the *free monoid* over X. The identity of this monoid is the empty sequence, which is denoted by ε and called the *empty word*. The number of letters that appear in $u \in X^*$ is denoted by $|u|$ and is called the *length* of the word u. In particular, $|\varepsilon| = 0$. For any non-empty set S, a function $f : X^* \to S$ is called a *word function*. If the set S is ordered, with an ordering $\leqslant$, then word functions can also be ordered pointwise, that is, for two word functions $f, g : X^* \to S$ we set that $f \leqslant g$ if $f(u) \leqslant g(u)$, for every word $u \in X^*$.

Let $\mathbb{S}$ be an arbitrary semiring and X an arbitrary alphabet. A *weighted finite automaton* over $\mathbb{S}$ and X is defined as a quadruple $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$, where:

(A1) $n \in \mathbb{N}$ is a natural number, called the *dimension* (or the *number of states*) of $\mathcal{A}$;
(A2) $\sigma^A \in \mathbb{S}^{1\times n}$ is a row vector, called the *initial weights vector* of $\mathcal{A}$;
(A3) $\{M_x^A\}_{x\in X} \subset \mathbb{S}^{n\times n}$ is a family of matrices indexed by X, which are called the *basic transition matrices* of $\mathcal{A}$;
(A4) $\tau^A \in \mathbb{S}^{n\times 1}$ is a column vector, called the *final weights vector* of $\mathcal{A}$.

The *composite transition matrix* M_u^A corresponding to a non-empty word $u = x_1x_2\dots x_k \in X^+$, where $x_1, x_2, \dots, x_k \in X$, is defined by

$$M_u^A = M_{x_1}^A \otimes M_{x_2}^A \otimes \cdots \otimes M_{x_k}^A,$$

and the transition matrix M_ε^A corresponding to the empty word ε is defined to be the identity matrix I_n of order n.

The behavior of the weighted finite automaton $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$ is a word function $[\![\mathcal{A}]\!] : X^* \to \mathbb{S}$ defined as follows: for a non-empty word $u = x_1x_2\dots x_k \in X^+$, where $x_1, x_2, \dots, x_k \in X$, we set

$$[\![\mathcal{A}]\!](u) = \sigma^A \otimes M_u^A \otimes \tau^A = \sigma^A \otimes M_{x_1}^A \otimes M_{x_2}^A \otimes \cdots \otimes M_{x_k}^A \otimes \tau^A, \tag{15}$$

and for the empty word ε we set

$$[\![\mathcal{A}]\!](\varepsilon) = \sigma^A \otimes M_\varepsilon^A \otimes \tau^A = \sigma^A \otimes \tau^A. \tag{16}$$

We say that $[\![\mathcal{A}]\!]$ is a *word function computed by* $\mathcal{A}$. In the literature, the terms *word function accepted by* $\mathcal{A}$ and *word function recognized by* $\mathcal{A}$ are also used. If $[\![\mathcal{A}]\!]$ is the zero function, i.e., $[\![\mathcal{A}]\!](u) = -\infty$ for each $u \in X^*$, we say that the weighted finite automaton $\mathcal{A}$ has *zero behavior*. Otherwise, $\mathcal{A}$ is said to have *non-zero behavior*.

For a weighted finite automaton $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$, the *reverse automaton* of $\mathcal{A}$ is the automaton

$$\overline{\mathcal{A}} = (n, \overline{\sigma}^A, \{\overline{M}_x^A\}_{x\in X}, \overline{\tau}^A),$$

where $\overline{\sigma}^A = \tau^A$, $\overline{\tau}^A = \sigma^A$ and $\overline{M}_x^A = (M_x^A)^\top$, for every $x \in X$.

In this paper, we deal with weighted finite automata that take their weights in the complete max-plus semiring $\mathbb{R}_\infty$. We will refer to such automata as *max-plus automata*. We will also use the same name for automata whose weights are taken in certain related semirings, such as $\mathbb{R}_{\max}$ and $\mathbb{Z}_\infty$, where we will specify the semiring in question.

Let $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$ and $\mathcal{B} = (m, \sigma^B, \{M_x^B\}_{x\in X}, \tau^B)$ be two max-plus automata over $\mathbb{R}_\infty$ and an alphabet X. If a matrix $U \in \mathbb{R}_\infty^{n\times m}$ satisfies the conditions

$$\sigma^A \leqslant \sigma^B \otimes U^\top, \tag{fs1}$$

$$U^\top \otimes M_x^A \leqslant M_x^B \otimes U^\top, \quad \text{for all } x \in X, \tag{fs2}$$

$$U^\top \otimes \tau^A \leqslant \tau^B, \tag{fs3}$$

then it is called a *forward simulation* between $\mathcal{A}$ and $\mathcal{B}$, and if it satisfies (fs2) and (fs3), then it is called a *forward presimulation* between $\mathcal{A}$ and $\mathcal{B}$.

If both U and $U^\top$ are forward simulations, that is, if the following is true

$$\sigma^A \leqslant \sigma^B \otimes U^\top, \ \ \sigma^B \leqslant \sigma^A \otimes U, \tag{fb1}$$

$$U^\top \otimes M_x^A \leqslant M_x^B \otimes U^\top, \ \ U \otimes M_x^B \leqslant M_x^A \otimes U, \quad \text{for all } x \in X, \tag{fb2}$$

$$U^\top \otimes \tau^A \leqslant \tau^B, \ \ U \otimes \tau^B \leqslant \tau^A \tag{fb3}$$

then U is called a *forward bisimulation* between $\mathcal{A}$ and $\mathcal{B}$, and if the conditions (fb2) and (fb3) are satisfied, then U is called a *forward prebisimulation* between $\mathcal{A}$ and $\mathcal{B}$.

On the other hand, a *backward simulation* between $\mathcal{A}$ and $\mathcal{B}$ is defined as a forward simulation between the reverse automata of $\mathcal{A}$ and $\mathcal{B}$, a *backward bisimulation* between $\mathcal{A}$ and $\mathcal{B}$ is defined as a forward bisimulation between their reverse automata, and in a similar way *backward presimulations* and *backward prebisimulations* are defined.

There are also two kinds of mixed-type bisimulations: *forward-backward bisimulations*, where U is a forward simulation and $U^\top$ is a backward simulation, and *backward-forward bisimulations*, where U is a backward simulation and $U^\top$ is a forward simulation. In a similar manner, we define *forward-backward presimulations* and *backward-forward presimulations*. The above-mentioned concepts of simulations and bisimulations were originally introduced in the context of fuzzy automata in [6, 7] (see also [8]), and were recently extended to the setting of max-plus automata in [9]. The terms presimulation and prebisimulation were proposed in [41, 42] in the context of fuzzy modal logics.

What is common to both types of simulations and all four types of bisimulations, as well as to the corresponding types of presimulations and prebisimulations, is that they can be viewed as solutions of certain systems of matrix inequalities. Namely, the above conditions (fs1)-(fs3), which define forward simulations, and (fb1)-(fb3), which define forward bisimulations, can be regarded as systems of matrix inequalities with the unknown matrix U. The systems of matrix inequalities that define the other types of simulations and bisimulations can be found in [9]. The procedures for solving these systems of inequalities were proposed in [9], and here we will explain them using the system (fs1)-(fs3) as an example (the procedures for solving the other systems are similar).

The part of the system (fs1)–(fs3) that determines forward presimulations, consisting of the inequalities from (fs2) and (fs3), is always solvable and has a greatest solution – the *greatest forward presimulation*. The procedure for computing the greatest forward presimulation consists of forming a non-increasing sequence of matrices $\{U_s\}_{s\in\mathbb{N}}$, which is defined inductively as follows:

$$U_1 = \tau^A \backslash \tau^B = (\tau^B / \tau^A)^\top, \qquad U_{s+1} = U_s \wedge \Big(\bigwedge_{x\in X} \big((M_x^B \otimes U_s^\top) / M_x^A \big)^\top \Big). \tag{17}$$

The greatest forward presimulation between $\mathcal{A}$ and $\mathcal{B}$ is the infimum of this sequence, and if it is also a solution of inequality (fs1), then it is the greatest forward simulation between $\mathcal{A}$ and $\mathcal{B}$. If it is not a solution of inequality (fs1), then no forward simulation exists between $\mathcal{A}$ and $\mathcal{B}$.

The greatest forward presimulation can be efficiently computed when the sequence stabilizes at some U_s, that is, when $U_s = U_{s+1}$ for some natural number s, and in that case, the greatest forward presimulation is equal to U_s. However, it may also happen that the sequence $\{U_s\}_{s\in\mathbb{N}}$ does not stabilize, i.e., that it is infinite, and in that case the greatest forward presimulation cannot be computed in a finite number of iterations.

Weak simulations and bisimulations, which we introduce in the next section generalizations of ordinary simulations and bisimulations, are also defined as solutions of certain systems of matrix inequalities. However, these systems are slightly different and are solved in a different manner.

4. Weak simulations and bisimulations for max-plus automata

Let $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$ be a max-plus automaton over the complete max-plus semiring $\mathbb{R}_\infty$ and an alphabet X. For an arbitrary word $u \in X^*$ we define vectors $\sigma_u^A \in \mathbb{R}_\infty^{1\times n}$ and $\tau_u^A \in \mathbb{R}_\infty^{n\times 1}$ as follows:

$$\sigma_u^A = \sigma^A \otimes M_u^A, \qquad \tau_u^A = M_u^A \otimes \tau^A. \tag{18}$$

The vectors σ_u^A ($u \in X^*$) will be called the *σ-vectors*, and the vectors τ_u^A ($u \in X^*$) will be called the *τ-vectors*. In the context of max-plus automata, σ-vectors are known as *generalized daters* (cf. [18, 30]) or *state vectors* (cf. [12, 31]). Both σ-vectors and τ-vectors play a very important role in the determinization of max-plus automata.

Further, let $\mathcal{A} = (n, \sigma^A, \{D_x^A\}_{x\in X}, \tau^A)$ and $\mathcal{B} = (m, \sigma^B, \{D_x^B\}_{x\in X}, \tau^B)$ be max-plus automata over the complete max-plus semiring $\mathbb{R}_\infty$ and an alphabet X.

A matrix $U \in R_\infty^{n\times m}$ is called a *weak forward simulation* between $\mathcal{A}$ and $\mathcal{B}$ if it satisfies the conditions

$$\sigma^A \leqslant \sigma^B \otimes U^\top, \tag{wfs1}$$
$$U^\top \otimes \tau_u^A \leqslant \tau_u^B, \ \text{ for all } u \in X^*, \tag{wfs2}$$

and if it satisfies (wfs2) but not necessarily (wfs1), it is called a *weak forward presimulation* between $\mathcal{A}$ and $\mathcal{B}$. If both U and $U^\top$ are weak forward simulations, then U is said to be a *weak forward bisimulation* between $\mathcal{A}$ and $\mathcal{B}$, and if both U and $U^\top$ are weak forward presimulations, then U is a *weak forward prebisimulation* between $\mathcal{A}$ and $\mathcal{B}$. In other words, U is a weak forward bisimulation between $\mathcal{A}$ and $\mathcal{B}$ if it satisfies the conditions

$$\sigma^A \leqslant \sigma^B \otimes U^\top, \ \ \sigma^B \leqslant \sigma^A \otimes U, \tag{wfb1}$$
$$U^\top \otimes \tau_u^A \leqslant \tau_u^B, \ \ U \otimes \tau_u^B \leqslant \tau_u^A, \ \text{ for all } u \in X^*, \tag{wfb2}$$

and it is a weak forward prebisimulation between $\mathcal{A}$ and $\mathcal{B}$ if it satisfies (wfb2) but not necessarily (wfb1).

On the other hand, if U satisfies the conditions

$$\tau^A \leqslant U \otimes \tau^B, \tag{wbs1}$$
$$\sigma_u^A \otimes U \leqslant \sigma_u^B, \ \text{ for all } u \in X^*, \tag{wbs2}$$

then it is called a *weak backward simulation* between $\mathcal{A}$ and $\mathcal{B}$, and if it satisfies (wbs2) but not necessarily (wbs1), then it is called a *weak backward presimulation* between $\mathcal{A}$ and $\mathcal{B}$. A *weak backward bisimulation* is a matrix U such that both U and $U^\top$ are weak backward simulations, while a *weak backward prebisimulation* is a matrix U such that both U and $U^\top$ are weak backward presimulations. More precisely, U is a weak backward bisimulation between $\mathcal{A}$ and $\mathcal{B}$ if it satisfies the conditions

$$\tau^A \leqslant U \otimes \tau^B, \;\; \tau^B \leqslant U^\top, \tag{wbb1}$$

$$\sigma^A_u \otimes U \leqslant \sigma^B_u, \;\; \sigma^B_u \otimes U^\top \leqslant \sigma^A_u, \;\; \text{for all } u \in X^*, \tag{wbb2}$$

and it is a weak backward prebisimulation between $\mathcal{A}$ and $\mathcal{B}$ if it satisfies (wbb2) but not necessarily (wbb1).

Just like the conditions that define ordinary simulations and bisimulations, the conditions that define weak simulations and bisimulations can also be treated as systems of matrix inequalities with the unknown matrix U.

The following theorem shows that weak simulations and bisimulations are generalizations of ordinary simulations and bisimulations, that is, the classes of weak simulations and bisimulations contain the corresponding classes of ordinary simulations and bisimulations as their subclasses. In other words, the theorem shows that the class of solutions to the system defining a given type of weak simulations and bisimulations is broader than the class of solutions to the system defining the corresponding type of ordinary simulations and bisimulations.

Theorem 4.1. *Let $\mathcal{A} = (n, \sigma^A, \{M^A_x\}_{x\in X}, \tau^A)$ and $\mathcal{B} = (m, \sigma^B, \{M^B_x\}_{x\in X}, \tau^B)$ be max-plus automata over $\mathbb{R}_\infty$ and an alphabet X. Then the following statements are true:*

a) *Every forward (resp. backward) presimulation between $\mathcal{A}$ and $\mathcal{B}$ is a weak forward (resp. backward) presimulation between $\mathcal{A}$ and $\mathcal{B}$.*
Accordingly, every forward (resp. backward) simulation between $\mathcal{A}$ and $\mathcal{B}$ is a weak forward (resp. backward) simulation between $\mathcal{A}$ and $\mathcal{B}$.
b) *Every forward (resp. backward) prebisimulation between $\mathcal{A}$ and $\mathcal{B}$ is a weak forward (resp. backward) prebisimulation between $\mathcal{A}$ and $\mathcal{B}$.*
Accordingly, every forward (resp. backward) bisimulation between $\mathcal{A}$ and $\mathcal{B}$ is a weak forward (resp. backward) bisimulation between $\mathcal{A}$ and $\mathcal{B}$.

Proof. It is sufficient to prove the statement concerning ordinary and weak forward presimulations. The corresponding statement for ordinary and weak forward simulations follows immediately, since ordinary forward presimulations and simulations differ only in condition (fs1), while weak forward presimulations and simulations differ only in condition (wfs1), which are identical. All other statements are proved in a similar way.

Let U be a forward presimulation, that is, let it satisfy (fs2) and (fs3). By induction on the length of the word u, we will prove that (wfs2) holds for every word $u \in X^*$. From (fs3) it follows directly that this holds for the empty word ε. Further, suppose that

$$U^\top \otimes \tau^A_u \leqslant \tau^B_u \tag{19}$$

holds for every word $u \in X^*$ such that $|u| = n$, and consider an arbitrary word $v \in X^*$ such that $|v| = n+1$. Then $v = xu$, for some $x \in X$ and $u \in X^*$ such that $|u| = n$, and according to (fs2) and (19) we obtain that

$$U^\top \otimes \tau^A_v = U^\top \otimes M^A_x \otimes \tau^A_u \leqslant M^B_x \otimes U^\top \otimes \tau^A_u \leqslant M^B_x \otimes \tau^B_u = \tau^B_v,$$

which completes our proof by mathematical induction. Therefore, we have proved that U is indeed a weak forward presimulation. □

Two max-plus automata $\mathcal{A}$ and $\mathcal{B}$ over $\mathbb{R}_\infty$ and an alphabet X are said to be *equivalent* if $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$. In that case, we also say that there is an *equivalence relation* between $\mathcal{A}$ and $\mathcal{B}$. If $[\![\mathcal{A}]\!] \leqslant [\![\mathcal{B}]\!]$, in the sense that $[\![\mathcal{A}]\!](u) \leqslant [\![\mathcal{B}]\!](u)$, for every word $u \in X^*$, then we say that there is a *containment relation* between $\mathcal{A}$ and $\mathcal{B}$.

As noted in the introduction, proving the existence of a simulation between two max-plus automata provides a certificate of the existence of a containment relation between them, whereas proving the existence of a bisimulation provides a certificate of the existence of equivalence. For this reason, detecting containment is the primary purpose of simulations, while detecting equivalence is the primary purpose of bisimulations. The following theorem shows that weak simulations and bisimulations can also be used for the same purposes, and, in light of Theorem 4.1 and Example 5.4, they are even more effective than ordinary simulations and bisimulations in doing so.

Theorem 4.2. *Let $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$ and $\mathcal{B} = (m, \sigma^B, \{M_x^B\}_{x\in X}, \tau^B)$ be max-plus automata over $\mathbb{R}_\infty$ and an alphabet X. Then the following statements are true:*

(a) *The existence of a weak simulation of any type between $\mathcal{A}$ and $\mathcal{B}$ provides confirmation of the existence of a containment relation between $\mathcal{A}$ and $\mathcal{B}$.*
(b) *The existence of a weak bisimulation of any type between $\mathcal{A}$ and $\mathcal{B}$ provides confirmation of the existence of an equivalence relation between $\mathcal{A}$ and $\mathcal{B}$.*

Proof. (a) Let there exists a weak forward simulation U between $\mathcal{A}$ and $\mathcal{B}$. Then, according to (wfs1) and (wfs2), for an arbitrary $u \in X^*$ we obtain that

$$[\![\mathcal{A}]\!](u) = \sigma^A \otimes M_u^A \otimes \tau^A = \sigma^A \otimes \tau_u^A \leqslant \sigma^B \otimes U^\top \otimes \tau_u^A \leqslant \sigma^B \otimes \tau_u^B = \sigma^B \otimes M_u^B \otimes \tau^B = [\![\mathcal{B}]\!](u),$$

and therefore, $[\![\mathcal{A}]\!] \leqslant [\![\mathcal{B}]\!]$. This completes the proof of statement (a) for weak forward simulations.
The statement in (a) concerning weak backward simulations is proved in a similar way.
(b) This follows directly from (a). □

The proofs of the next two theorems are not very complicated and will be omitted. The first of these theorems establishes a fundamental connection between weak forward and weak backward simulations and bisimulations, while the second concerns the closure of the considered types of weak simulations and bisimulations under matrix multiplication, addition, and transposition.

Theorem 4.3. *Let $\mathcal{A}$ and $\mathcal{B}$ be max-plus automata over $\mathbb{R}_\infty$ and an alphabet X. Then the following statements are true:*

(a) *A matrix $U \in \mathbb{R}_\infty^{n\times m}$ is a weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$ if and only if it is a weak backward simulation between the reverse automata of $\mathcal{A}$ and $\mathcal{B}$.*
(b) *A matrix $U \in \mathbb{R}_\infty^{n\times m}$ is a weak forward bisimulation between $\mathcal{A}$ and $\mathcal{B}$ if and only if it is a weak backward bisimulation between the reverse automata of $\mathcal{A}$ and $\mathcal{B}$.*

Theorem 4.4. *Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be max-plus automata over $\mathbb{R}_\infty$ and an alphabet X. Then the following statements are true:*

(a) *The product of a weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$ and a weak forward simulation between $\mathcal{B}$ and $\mathcal{C}$ is a weak forward simulation between $\mathcal{A}$ and $\mathcal{C}$.*
(b) *The sum of an arbitrary family of weak forward simulations between $\mathcal{A}$ and $\mathcal{B}$ is a weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$.*

Statements (a) and (b) remain valid when the term "forward" is replaced by "backward", and/or when the term "simulation" is replaced by "bisimulation."
In addition, the following also holds:

(c) *The transpose of a weak forward (resp. backward) bisimulation between $\mathcal{A}$ and $\mathcal{B}$ is a weak forward (resp. backward) bisimulation between $\mathcal{B}$ and $\mathcal{A}$.*

The following theorem is the main result of the paper. It provides a procedure for testing the existence of weak simulations and bisimulations, as well as for computing the greatest weak simulations and bisimulations in cases where they exist.

Theorem 4.5. *Let $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$ and $\mathcal{B} = (m, \sigma^B, \{M_x^B\}_{x\in X}, \tau^B)$ be max-plus automata over $\mathbb{R}_\infty$ and an alphabet X, and let*

$$U^{\mathrm{wfs}} = \bigwedge_{u\in X^*} \tau_u^A \backslash \tau_u^B, \quad U^{\mathrm{wbs}} = \bigwedge_{u\in X^*} \sigma_u^A \backslash \sigma_u^B, \quad U^{\mathrm{wfb}} = \bigwedge_{u\in X^*} \tau_u^A \mid \tau_u^B, \quad U^{\mathrm{wbb}} = \bigwedge_{u\in X^*} \sigma_u^A \mid \sigma_u^B.$$

Then the following statements are true:

(a) U^{wfs} *is the greatest weak forward presimulation between* $\mathcal{A}$ *and* $\mathcal{B}$.
If U^{wfs} *satisfies* (wfs1)*, then it is the greatest weak forward simulation between* $\mathcal{A}$ *and* $\mathcal{B}$*, and if it does not satisfy* (wfs1)*, then there exists no weak forward simulation between* $\mathcal{A}$ *and* $\mathcal{B}$.

(b) U^{wbs} *is the greatest weak backward presimulation between* $\mathcal{A}$ *and* $\mathcal{B}$.
If U^{wbs} *satisfies* (wbs1)*, then it is the greatest weak backward simulation between* $\mathcal{A}$ *and* $\mathcal{B}$*, and if it does not satisfy* (wbs1)*, then there exists no weak backward simulation between* $\mathcal{A}$ *and* $\mathcal{B}$.

(c) U^{wfb} *is the greatest weak forward prebisimulation between* $\mathcal{A}$ *and* $\mathcal{B}$.
If U^{wfb} *satisfies* (wfb1)*, then it is the greatest weak forward bisimulation between* $\mathcal{A}$ *and* $\mathcal{B}$*, and if it does not satisfy* (wfb1)*, then there exists no weak forward bisimulation between* $\mathcal{A}$ *and* $\mathcal{B}$.

(d) U^{wbb} *is the greatest weak backward prebisimulation between* $\mathcal{A}$ *and* $\mathcal{B}$.
If U^{wbb} *satisfies* (wbs1)*, then it is the greatest weak backward bisimulation between* $\mathcal{A}$ *and* $\mathcal{B}$*, and if it does not satisfy* (wbb1)*, then there exists no weak backward bisimulation between* $\mathcal{A}$ *and* $\mathcal{B}$.

Proof. We will prove only statement (a). The remaining statements are proved in a similar way.

(a) According to the residuation property (11), a matrix $U \in \mathbb{R}_\infty^{n\times m}$ satisfies (wfs2) if and only if

$$U \leqslant \tau_u^A \between \tau_u^B, \quad \text{for every } u \in X^*,$$

and hence, U^{wfs} is the greatest solution of (wfs2), i.e., it is the greatest weak forward presimulation between $\mathcal{A}$ and $\mathcal{B}$.

Assume that U^{wfs} satisfies (wfs1). Then it is clear that U^{wfs} is a weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$, and since any weak forward simulation U between $\mathcal{A}$ and $\mathcal{B}$ is also a weak forward presimulation, and therefore $U \leqslant U^{\mathrm{wfs}}$, we conclude that U^{wfs} is the greatest weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$.

Next, suppose that U^{wfs} does not satisfy (wfs1). If there exists a weak forward simulation U between $\mathcal{A}$ and $\mathcal{B}$, then U is also a weak forward presimulation, whence it follows that $U \leqslant U^{\mathrm{wfs}}$. Since U satisfies (wfs1), we obtain that

$$\sigma^A \leqslant \sigma^B \otimes U^\top \leqslant \sigma^B \otimes (U^{\mathrm{wfs}})^\top,$$

which contradicts the assumption that U^{wfs} does not satisfy (wfs1). Therefore, we conclude that if U^{wfs} does not satisfy (wfs1), then there exists no weak forward simulation between between $\mathcal{A}$ and $\mathcal{B}$. □

According to the previous theorem, when testing the existence of a weak simulation or weak bisimulation of a given type, the main difficulty lies in computing the greatest weak presimulation or the greatest weak prebisimulation of that type. Once this greatest weak presimulation or weak prebisimulation has been computed, it is straightforward to verify whether it satisfies the additional conditions required to qualify as a weak simulation or weak bisimulation of the corresponding type. The problem of computing the greatest weak presimulations and the greatest weak prebisimulations will be addressed in the next section.

5. Computational procedure

In this section, we describe a procedure for testing the existence of a weak simulation/bisimulation of a given type, and for computing the greatest such simulation/bisimulation in cases where it exists. In fact, by this procedure, one first computes the greatest weak presimulation/prebisimulation of the given type, and then checks whether it is a weak simulation/bisimulation of that type. We also provide computational examples illustrating the application of this procedure, as well as cases in which the procedure cannot be applied.

Procedure 1 Computing the greatest weak simulation/bisimulation between max-plus automata

Input: Max-plus automata $\mathcal{A} = (n, \sigma^A, \{M_x^A\}_{x\in X}, \tau^A)$ and $\mathcal{B} = (m, \sigma^B, \{M_x^B\}_{x\in X}, \tau^B)$;
input alphabet X;
simType $\in$ {wfs, wbs, wfb, wbb} (specifies the type of a weak simulation/bisimulation).

Output: The greatest weak simulation/bisimulation U, if it exists, or null

Initialize a queue *toProcess* with the initial pair (t^A, t^B) depending on *simType*:

$$toProcess = (t^A, t^B) = \begin{cases} (\tau^A, \tau^B), & \text{if } simType = \text{wfs or } simType = \text{wfb} \\ (\sigma^A, \sigma^B), & \text{if } simType = \text{wbs or } simType = \text{wbb} \end{cases}$$

Initialize the matrix U depending on *simType*:

$$U = \begin{cases} t^A \mathbin{\backslash\!\!\!\circ}\, t^B & \text{if } simType = \text{wfs or } simType = \text{wbs} \\ t^A \mathbin{|\!\!\!\circ}\, t^B & \text{if } simType = \text{wfb or } simType = \text{wbb} \end{cases}$$

while *toProcess* is not empty **do**
    $(t^A, t^B) = toProcess$.dequeue();
    for each $x \in X$ **do**

$$(t_*^A, t_*^B) = \begin{cases} (M_x^A \otimes t^A, M_x^B \otimes t^B) & \text{if } simType = \text{wfs or } simType = \text{wfb} \\ (t^A \otimes M_x^A, t^B \otimes M_x^B) & \text{if } simType = \text{wbs or } simType = \text{wbb} \end{cases}$$

        Enqueue (t_*^A, t_*^B) into *toProcess* if it has not been processed yet

$$U = \begin{cases} U \wedge (t_*^A \mathbin{\backslash\!\!\!\circ}\, t_*^B) & \text{if } simType = \text{wfs or } simType = \text{wbs} \\ U \wedge (t_*^A \mathbin{|\!\!\!\circ}\, t_*^B) & \text{if } simType = \text{wfb or } simType = \text{wbb} \end{cases}$$

    end for
end while
Check the initial condition depending on the selected type:

$$condition = \begin{cases} \sigma^A \leqslant \sigma^B \otimes U^T & \text{if } simType = \text{wfs} \\ \tau^A \leqslant U \otimes \tau^B & \text{if } simType = \text{wbs} \\ \sigma^A \leqslant \sigma^B \otimes U^\top \text{ and } \sigma^B \leqslant \sigma^A \otimes U & \text{if } simType = \text{wfb} \\ \tau^A \leqslant U \otimes \tau^B \text{ and } \tau^B \leqslant U^\top \otimes \tau^A & \text{if } simType = \text{wbb} \end{cases}$$

if *condition* **then**
    return U
else
    return null
end if

We will explain the application of this procedure in the case *simType* = wfs. In the remaining cases, the procedure is very similar. The procedure begins by initializing the *toProcess* queue with the initial pair (τ^A, τ^B), and by initializing the matrix U, which the procedure computes, by setting $U = \tau^A \mathbin{\backslash\!\!\!\circ}\, \tau^B$. When the τ-pair (t^A, t^B) is removed from the top of the *toProcess* queue by the dequeue operation, new τ-pairs $(M_x^A \otimes t^A, M_x^B \otimes t^B)$ are formed from it, one by one, for all $x \in X$. After each of these pairs (t_*^A, t_*^B) is formed, it is checked whether it is equal to any previously generated τ-pair, and only those τ-pairs that are different from all previously generated τ-pairs are inserted at the bottom of the queue by the enqueue operation. The previously generated distinct τ-pairs are stored in a separate memory. Immediately after the pair (t_*^A, t_*^B) is inserted into the queue, the residual $t_*^A \backslash t_*^B$ is computed, and the matrix U is replaced with its infimum with that residual. If the number of τ-pairs is finite, the main loop of the procedure will terminate once the *toProcess* queue becomes empty, i.e., when no new pairs are added to the queue. The matrix U obtained at the moment of termination is the greatest weak forward presimulation, and it is then checked whether it satisfies condition (fs1). If it does, the matrix U is returned as the greatest weak forward simulation, and if it does not, then *null* is returned as an indication that no weak forward simulation exists between the considered max-plus automata.

Although the iterative procedure for computing the greatest weak simulation or bisimulation does not guarantee termination in the general case, the computational cost of a single iteration can be estimated as follows. Let n and m denote the number of states of automata $\mathcal{A}$ and $\mathcal{B}$, respectively, and $l = |X|$ the size of the input alphabet. Each iteration processes one pair of vectors (t^A, t^B) and, for each input symbol $x \in X$, computes $t_*^A = M_x^A \otimes t^A$ and $t_*^B = M_x^B \otimes t^B$, followed by the update of the relation U. Given that $M_x^A \in \mathbb{R}_\infty^{n\times n}$, $M_x^B \in \mathbb{R}_\infty^{m\times m}$ and $U \in \mathbb{R}_\infty^{n\times m}$, the computational cost of processing a single letter is $O(n^2 + m^2 + nm)$. As all $l = |X|$ letters are processed in each iteration, the total complexity per processed pair is $O(l \cdot (n^2 + m^2 + nm)) = O\big(l \cdot \max(n^2, m^2)\big)$. Since the total number of generated pairs is unbounded in general, the overall procedure may not terminate, practical implementations typically should have a stopping criterion, such as a maximal number of iterations or a time limit.

The following examples illustrate the application of the above procedure and demonstrate several cases in which the procedure cannot be applied.

Example 5.1. *For the input alphabet $X = \{x, y\}$, let $\mathcal{A} = (2, \sigma^A, \{M_x^A, M_y^A\}, \tau^A)$ and $\mathcal{B} = (3, \sigma^B, \{M_x^B, M_y^B\}, \tau^B)$ be max-plus automata over X given by the following vectors and matrices:*

$$M_x^A = \begin{bmatrix} 0 & -\infty \\ -\infty & -\infty \end{bmatrix}, \qquad M_y^A = \begin{bmatrix} -\infty & 2 \\ -\infty & -\infty \end{bmatrix}, \qquad \sigma^A = \begin{bmatrix} 1 & 2 \end{bmatrix}, \qquad \tau^A = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

$$M_x^B = \begin{bmatrix} 0 & -\infty & -\infty \\ -\infty & -\infty & 1 \\ -\infty & -\infty & 0 \end{bmatrix}, \qquad M_y^B = \begin{bmatrix} -\infty & -\infty & 2 \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \end{bmatrix}, \qquad \sigma^B = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}, \qquad \tau^B = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}.$$

(a) There is a weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$: *The max-plus automaton $\mathcal{A}$ has four different τ-vectors*

$$\tau_\varepsilon^A = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \qquad \tau_{x^k}^A = \begin{bmatrix} 1 \\ -\infty \end{bmatrix}, \text{ for } k \in \mathbb{N}, \qquad \tau_{x^k y}^A = \begin{bmatrix} 0 \\ -\infty \end{bmatrix}, \text{ for } k \in \mathbb{N}_0, \qquad \tau_u^A = \begin{bmatrix} -\infty \\ -\infty \end{bmatrix}, \text{ for all other } u \in X^*,$$

while the max-plus automaton $\mathcal{B}$ has only two different τ-vectors

$$\tau_\varepsilon^B = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} = \tau_{y^k}^B, \text{ for } k \in \mathbb{N}, \qquad \tau_u^B = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \tau_{y^k}^B, \text{ for } u \in X^+ \text{ such that } |u|_x \neq 0,$$

and we wlll see that there are six different pairs of τ-vectors.

The procedure for computing the pairs of τ-vectors, the corresponding residuals, as well as the infima, which ultimately yield the greatest weak forward presimulation, carried out according to Procedure 1, is presented in Table 1. The symbol ⊠ written next to a pair of τ-vectors indicates that this pair is a repetition of a previously computed pair. Consequently, no new pair is computed from it, as it would also be a repetition of an earlier pair, nor its residual is computed, as it would be a repetition of a previously computed residual and would not affect the computation of the infimum of the family of residuals due to the idempotency and commutativity of the infimum operation.

As shown in Table 1, the infimum of the family of residuals of pairs of τ-vectors, that is, the greatest weak forward presimulation between $\mathcal{A}$ and $\mathcal{B}$, is the matrix

$$U = \begin{bmatrix} 1 & -2 & -1 \\ 4 & 1 & 2 \end{bmatrix}.$$

Table 1: Details of the computation of the greatest weak forward simulation in Example 5.1.

Step number	Word length	Pair of τ-vectors that has been computed	The residual of the computed pair	The infimum after the current step
1	0	$(\tau^A_\varepsilon, \tau^B_\varepsilon) = \left(\begin{bmatrix}1\\-2\end{bmatrix}, \begin{bmatrix}2\\-1\\0\end{bmatrix}\right)$	$\tau^A_\varepsilon \between \tau^B_\varepsilon = \begin{bmatrix}1 & -2 & -1\\4 & 1 & 2\end{bmatrix}$	$U := \begin{bmatrix}1 & -2 & -1\\4 & 1 & 2\end{bmatrix}$
2	1	$(\tau^A_x, \tau^B_x) = \left(\begin{bmatrix}1\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\tau^A_x \between \tau^B_x = \begin{bmatrix}1 & 0 & -1\\+\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}1 & -2 & -1\\4 & 1 & 2\end{bmatrix}$
3		$(\tau^A_y, \tau^B_y) = \left(\begin{bmatrix}0\\-\infty\end{bmatrix}, \begin{bmatrix}2\\-1\\0\end{bmatrix}\right)$	$\tau^A_y \between \tau^B_y = \begin{bmatrix}2 & -1 & 0\\+\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}1 & -2 & -1\\4 & 1 & 2\end{bmatrix}$
4	2	$(\tau^A_{x^2}, \tau^B_{x^2}) = \left(\begin{bmatrix}1\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\boxtimes$	
5		$(\tau^A_{xy}, \tau^B_{xy}) = \left(\begin{bmatrix}0\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\tau^A_{xy} \between \tau^B_{xy} = \begin{bmatrix}2 & 1 & 0\\+\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}1 & -2 & -1\\4 & 1 & 2\end{bmatrix}$
6		$(\tau^A_{yx}, \tau^B_{yx}) = \left(\begin{bmatrix}-\infty\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\tau^A_{yx} \between \tau^B_{yx} = \begin{bmatrix}+\infty & +\infty & +\infty\\+\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}1 & -2 & -1\\4 & 1 & 2\end{bmatrix}$
7		$(\tau^A_{y^2}, \tau^B_{y^2}) = \left(\begin{bmatrix}-\infty\\-\infty\end{bmatrix}, \begin{bmatrix}2\\-1\\0\end{bmatrix}\right)$	$\tau^A_{y^2} \between \tau^B_{y^2} = \begin{bmatrix}+\infty & +\infty & +\infty\\+\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}1 & -2 & -1\\4 & 1 & 2\end{bmatrix}$
8	3	$(\tau^A_{x^2y}, \tau^B_{x^2y}) = \left(\begin{bmatrix}0\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\boxtimes$	
9		$(\tau^A_{yxy}, \tau^B_{yxy}) = \left(\begin{bmatrix}-\infty\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\boxtimes$	
10		$(\tau^A_{xyx}, \tau^B_{xyx}) = \left(\begin{bmatrix}-\infty\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\boxtimes$	
11		$(\tau^A_{y^2x}, \tau^B_{y^2x}) = \left(\begin{bmatrix}-\infty\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\boxtimes$	
12		$(\tau^A_{xy^2}, \tau^B_{xy^2}) = \left(\begin{bmatrix}-\infty\\-\infty\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}\right)$	$\boxtimes$	
13		$(\tau^A_{y^3}, \tau^B_{y^3}) = \left(\begin{bmatrix}-\infty\\-\infty\end{bmatrix}, \begin{bmatrix}2\\-1\\0\end{bmatrix}\right)$	$\boxtimes$	

Since

$$\sigma^B \otimes U^\top = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 4 \\ -2 & 1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 4 \end{bmatrix} > \begin{bmatrix} 1 & 2 \end{bmatrix} = \sigma^A,$$

we conclude that U is the greatest weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$.

As can be seen from the table, the computation was carried out in 13 steps, where a single step consists of generating one pair of τ-vectors, computing the residual of that pair, and taking the infimum of this residual with the infimum obtained in the previous step. It is worth noting that the infimum of the family of pairs of τ-vectors, that is, the greatest weak forward presimulation, is attained already in the first step, and all subsequent computations might appear superfluous. However, at present we do not have any general procedure by which we could detect that the infimum has already been reached, which is why it was necessary to continue the computation until all pairs of τ-vectors were generated. The problem of detecting when the infimum has been reached will be one of the main topics of our future research.

(b) There is no a weak backward simulation between $\mathcal{A}$ and $\mathcal{B}$: *The procedure for computing the greatest weak backward presimulation between $\mathcal{A}$ and $\mathcal{B}$, carried out according to Procedure 1, is presented in Table 2.*

The greatest weak backward presimulation between $\mathcal{A}$ and $\mathcal{B}$, computed according to Procedure 1, is

$$U = \begin{bmatrix} -1 & -\infty & -2 \\ -\infty & -\infty & -3 \end{bmatrix},$$

and since

$$U \otimes \tau^B = \begin{bmatrix} -1 & -\infty & -2 \\ -\infty & -\infty & -3 \end{bmatrix} \otimes \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \end{bmatrix} \ngeq \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \tau^A,$$

we conclude that U does not satisfy the condition (wbs1), which means that there is no a weak backward simulation between $\mathcal{A}$ and $\mathcal{B}$.

Table 2: Details of the computation of the greatest weak backward presimulation in Example 5.1 (b).

Step number	Word length	Pair of τ-vectors that has been computed		The residual of the computed pair	The infimum after the current step
1	0	$(\sigma^A_\varepsilon, \sigma^B_\varepsilon) = \left(\begin{bmatrix}1 & 2\end{bmatrix}, \begin{bmatrix}0 & 1 & -1\end{bmatrix}\right)$		$\sigma^A_\varepsilon \backslash \sigma^B_\varepsilon = \begin{bmatrix}-1 & 0 & -2\\ -2 & -1 & -3\end{bmatrix}$	$U := \begin{bmatrix}-1 & 0 & -2\\ -2 & -1 & -3\end{bmatrix}$
2	1	$(\sigma^A_x, \sigma^B_x) = \left(\begin{bmatrix}1 & -\infty\end{bmatrix}, \begin{bmatrix}0 & -\infty & 2\end{bmatrix}\right)$		$\sigma^A_x \backslash \sigma^B_x = \begin{bmatrix}-1 & -\infty & 1\\ +\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}-1 & -\infty & -2\\ -2 & -1 & -3\end{bmatrix}$
3		$(\sigma^A_y, \sigma^B_y) = \left(\begin{bmatrix}-\infty & 3\end{bmatrix}, \begin{bmatrix}-\infty & 1 & 2\end{bmatrix}\right)$		$\sigma^A_y \backslash \sigma^B_y = \begin{bmatrix}+\infty & +\infty & +\infty\\ -\infty & -2 & -1\end{bmatrix}$	$U := \begin{bmatrix}-1 & -\infty & -2\\ -\infty & -2 & -3\end{bmatrix}$
4	2	$(\sigma^A_{x^2}, \sigma^B_{x^2}) = \left(\begin{bmatrix}1 & -\infty\end{bmatrix}, \begin{bmatrix}0 & -\infty & 2\end{bmatrix}\right)$	⊠		
5		$(\sigma^A_{xy}, \sigma^B_{xy}) = \left(\begin{bmatrix}-\infty & 3\end{bmatrix}, \begin{bmatrix}-\infty & -\infty & 2\end{bmatrix}\right)$		$\sigma^A_{xy} \backslash \sigma^B_{xy} = \begin{bmatrix}+\infty & +\infty & +\infty\\ -\infty & -\infty & -1\end{bmatrix}$	$U := \begin{bmatrix}-1 & -\infty & -2\\ -\infty & -\infty & -3\end{bmatrix}$
6		$(\sigma^A_{yx}, \sigma^B_{yx}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & -\infty & 2\end{bmatrix}\right)$		$\sigma^A_{yx} \backslash \sigma^B_{yx} = \begin{bmatrix}+\infty & +\infty & +\infty\\ +\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}-1 & -\infty & -2\\ -\infty & -\infty & -3\end{bmatrix}$
7		$(\sigma^A_{y^2}, \sigma^B_{y^2}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & 1 & 2\end{bmatrix}\right)$		$\sigma^A_{y^2} \backslash \sigma^B_{y^2} = \begin{bmatrix}+\infty & +\infty & +\infty\\ +\infty & +\infty & +\infty\end{bmatrix}$	$U := \begin{bmatrix}-1 & -\infty & -2\\ -\infty & -\infty & -3\end{bmatrix}$
8	3	$(\sigma^A_{xyx}, \sigma^B_{xyx}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & -\infty & 2\end{bmatrix}\right)$	⊠		
9		$(\sigma^A_{xy^2}, \sigma^B_{xy^2}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & -\infty & 2\end{bmatrix}\right)$	⊠		
10		$(\sigma^A_{yx^2}, \sigma^B_{yx^2}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & -\infty & 2\end{bmatrix}\right)$	⊠		
11		$(\sigma^A_{yxy}, \sigma^B_{yxy}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & -\infty & 2\end{bmatrix}\right)$	⊠		
12		$(\sigma^A_{y^2x}, \sigma^B_{y^2x}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & -\infty & 2\end{bmatrix}\right)$	⊠		
13		$(\sigma^A_{y^3}, \sigma^B_{y^3}) = \left(\begin{bmatrix}-\infty & -\infty\end{bmatrix}, \begin{bmatrix}-\infty & 1 & 2\end{bmatrix}\right)$	⊠		

In the previous example, we observed a situation in which a weak forward simulation exists, while no weak backward simulation does. It is easy to construct an example exhibiting the opposite situation, for instance, by considering the reverse automata of those used in the previous example. This demonstrates that weak forward and weak backward simulations are mutually independent. Consequently, from the perspective of their application in proving containment, neither of them can be given priority a priori. Instead, if one of them is shown not to exist, we simply proceed to search for the other.

The same observations apply to weak forward and weak backward bisimulations as well.

Example 5.2. *For the input alphabet $X = \{x, y\}$, let $\mathcal{A} = (2, \sigma^A, \{M^A_x, M^A_y\}, \tau^A)$ and $\mathcal{B} = (3, \sigma^B, \{M^B_x, M^B_y\}, \tau^B)$ be* *max-plus automata over X given by the following vectors and matrices:*

$$M_x^A = \begin{bmatrix} 0 & -\infty \\ -\infty & -\infty \end{bmatrix}, \qquad M_y^A = \begin{bmatrix} -\infty & 2 \\ -\infty & -\infty \end{bmatrix}, \qquad \sigma^A = \begin{bmatrix} 1 & 2 \end{bmatrix}, \qquad \tau^A = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

$$M_x^B = \begin{bmatrix} 0 & -\infty & -\infty \\ -\infty & -\infty & 1 \\ -\infty & -\infty & -1 \end{bmatrix}, \qquad M_y^B = \begin{bmatrix} -\infty & -\infty & 2 \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \end{bmatrix}, \qquad \sigma^B = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}, \qquad \tau^B = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}.$$

The max-plus automaton $\mathcal{A}$ is the same as the automaton $\mathcal{A}$ from Example 5.1 and has four different τ-vectors

$$\tau_\varepsilon^A = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \qquad \tau_{x^k}^A = \begin{bmatrix} 1 \\ -\infty \end{bmatrix}, \text{ for } k \in \mathbb{N}, \qquad \tau_{x^k y}^A = \begin{bmatrix} 0 \\ -\infty \end{bmatrix}, \text{ for } k \in \mathbb{N}_0, \qquad \tau_u^A = \begin{bmatrix} -\infty \\ -\infty \end{bmatrix}, \text{ for } u \in L,$$

where $L = X^* \setminus (\{x^k \mid k \in \mathbb{N}_0\} \cup \{x^k y \mid k \in \mathbb{N}_0\})$, while τ-vectors of the max-plus automaton $\mathcal{B}$ are given by

$$\tau_\varepsilon^B = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} = \tau_{y^k}^B, \text{ for } k \in \mathbb{N}, \qquad \tau_u^B = \begin{bmatrix} -|u|_x + \ell(u) + 2 \\ -|u|_x + 2 \\ -|u|_x \end{bmatrix}, \text{ for } u \in X^+ \text{ such that } |u|_x \neq 0,$$

where $|u|_x$ denotes the number of occurrences of the letter x in the word u, while $\ell(u)$ denotes the length of the longest prefix of u that does not contain y, that is, $\ell(u) = k$ if $u = x^k$ or $u = x^k yv$, for some $v \in X^*$. Therefore, there exist infinitely many τ-vectors in $\mathcal{B}$, and consequently infinitely many pairs of τ-vectors (τ_u^A, τ_u^B). The pairs of τ-vectors and the corresponding residuals are shown in Table 3.

Table 3: Pairs of τ-vectors computed in Example 5.2 and their residuals.

Pair of τ-vectors	**The residual of the pair of τ-vectors**	
$(\tau_\varepsilon^A, \tau_\varepsilon^B) = \left(\begin{bmatrix} 1 \\ -2 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}\right)$	$\tau_\varepsilon^A \between \tau_\varepsilon^B = \begin{bmatrix} 1 & -1 & -1 \\ 4 & 2 & 2 \end{bmatrix}$	
$(\tau_y^A, \tau_y^B) = \left(\begin{bmatrix} 0 \\ -\infty \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}\right)$	$\tau_y^A \between \tau_y^B = \begin{bmatrix} 2 & 0 & 0 \\ +\infty & +\infty & +\infty \end{bmatrix}$	
$(\tau_{x^k}^A, \tau_{x^k}^B) = \left(\begin{bmatrix} 1 \\ -\infty \end{bmatrix}, \begin{bmatrix} 2 \\ -k+2 \\ -k \end{bmatrix}\right)$	$\tau_{x^k}^A \between \tau_{x^k}^B = \begin{bmatrix} 1 & -k+1 & -k-1 \\ +\infty & +\infty & +\infty \end{bmatrix}$	$(k \in \mathbb{N})$
$(\tau_{x^k y}^A, \tau_{x^k y}^B) = \left(\begin{bmatrix} 0 \\ -\infty \end{bmatrix}, \begin{bmatrix} 2 \\ -k+2 \\ -k \end{bmatrix}\right)$	$\tau_{x^k}^A \between \tau_{x^k}^B = \begin{bmatrix} 2 & -k+2 & -k \\ +\infty & +\infty & +\infty \end{bmatrix}$	$(k \in \mathbb{N})$
$(\tau_u^A, \tau_u^B) = \left(\begin{bmatrix} -\infty \\ -\infty \end{bmatrix}, \tau_u^B\right)$	$\tau_u^A \between \tau_u^B = \begin{bmatrix} +\infty & +\infty & +\infty \\ +\infty & +\infty & +\infty \end{bmatrix}$	$(u \in L)$

Therefore, the greatest weak forward presimulation between $\mathcal{A}$ and $\mathcal{B}$ is given by

$$U = \bigwedge_{u \in X^*} \tau_u^A \between \tau_u^B = \begin{bmatrix} 1 & -1 & -1 \\ 4 & 2 & 2 \end{bmatrix} \wedge \begin{bmatrix} 2 & 0 & 0 \\ +\infty & +\infty & +\infty \end{bmatrix} \wedge \left(\bigwedge_{k \in \mathbb{N}} \begin{bmatrix} 1 & -k+1 & -k-1 \\ +\infty & +\infty & +\infty \end{bmatrix} \right)$$

$$\wedge \left(\bigwedge_{k \in \mathbb{N}} \begin{bmatrix} 2 & -k+2 & -k \\ +\infty & +\infty & +\infty \end{bmatrix} \right) \wedge \begin{bmatrix} +\infty & +\infty & +\infty \\ +\infty & +\infty & +\infty \end{bmatrix} = \begin{bmatrix} 1 & -\infty & -\infty \\ 4 & 2 & 2 \end{bmatrix},$$

and since

$$\sigma^B \otimes U^\top = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 4 \\ -\infty & 2 \\ -\infty & 2 \end{bmatrix} = \begin{bmatrix} 1 & 4 \end{bmatrix} > \begin{bmatrix} 1 & 2 \end{bmatrix} = \sigma^A,$$

we obtain that U is the greatest weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$.

However, since the number of pairs of τ-vectors is infinite, the greatest weak forward simulation cannot be computed efficiently in the way proposed by Procedure 1.

Example 5.3. *For the input alphabet $X = \{x, y\}$, let $\mathcal{A} = (2, \sigma^A, \{M_x^A, M_y^A\}, \tau^A)$ and $\mathcal{B} = (3, \sigma^B, \{M_x^B, M_y^B\}, \tau^B)$ be max-plus automata over X given by the following vectors and matrices:*

$$M_x^A = \begin{bmatrix} 4 & 1 \\ -\infty & 2 \end{bmatrix}, \qquad M_y^A = \begin{bmatrix} -\infty & -\infty \\ -\infty & 1 \end{bmatrix}, \qquad \sigma^A = \begin{bmatrix} 1 & -\infty \end{bmatrix}, \qquad \tau^A = \begin{bmatrix} -\infty \\ 0 \end{bmatrix},$$

$$M_x^B = \begin{bmatrix} -\infty & 4 & -\infty \\ -\infty & 4 & -\infty \\ -\infty & -\infty & 2 \end{bmatrix}, \qquad M_y^B = \begin{bmatrix} -\infty & -\infty & 0 \\ -\infty & -\infty & -2 \\ -\infty & -\infty & 1 \end{bmatrix}, \qquad \sigma^B = \begin{bmatrix} 1 & -\infty & -\infty \end{bmatrix}, \qquad \tau^B = \begin{bmatrix} -1 \\ -3 \\ 0 \end{bmatrix}.$$

These automata were already considered in Example 4.2 of [9], where the existence of forward simulations was tested and the greatest forward simulation between them was computed. Here, we will do the same for weak forward simulations.

Both automata have infinitely many τ-vectors, and therefore there are also infinitely many pairs of τ-vectors which, together with the corresponding residuals, are presented in Table 4.

Table 4: Pairs of τ-vectors computed in Example 5.3 and their residuals.

Pair of τ-vectors	**The residual of the pair of τ-vectors**	
$(\tau_\varepsilon^A, \tau_\varepsilon^B) = \left(\begin{bmatrix} -\infty \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ -3 \\ 0 \end{bmatrix}\right)$	$\tau_\varepsilon^A \backslash \tau_\varepsilon^B = \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & -3 & 0 \end{bmatrix}$	
$(\tau_u^A, \tau_u^B) = \left(\begin{bmatrix} -\infty \\ 2\lvert u\rvert_x + \lvert u\rvert_y \end{bmatrix}, \begin{bmatrix} 2\lvert u\rvert_x + \lvert u\rvert_y - 1 \\ 2\lvert u\rvert_x + \lvert u\rvert_y - 3 \\ 2\lvert u\rvert_x + \lvert u\rvert_y \end{bmatrix}\right)$	$\tau_u^A \backslash \tau_u^B = \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & -3 & 0 \end{bmatrix}$	$(h(u) = y)$
$(\tau_u^A, \tau_u^B) = \left(\begin{bmatrix} 2\lvert u\rvert_x + \lvert u\rvert_y + 2\ell(u) - 3 \\ 2\lvert u\rvert_x + \lvert u\rvert_y \end{bmatrix}, \begin{bmatrix} 2\lvert u\rvert_x + \lvert u\rvert_y + 2\ell(u) - 3 \\ 2\lvert u\rvert_x + \lvert u\rvert_y + 2\ell(u) - 3 \\ 2\lvert u\rvert_x + \lvert u\rvert_y \end{bmatrix}\right)$	$\tau_u^A \backslash \tau_u^B = \begin{bmatrix} 0 & 0 & -2\ell(u) + 3 \\ 2\ell(u) - 3 & 2\ell(u) - 3 & 0 \end{bmatrix}$	$(h(u) = x)$

Recall that $|u|_x$ and $|u|_y$ denote respectively the numbers of occurrences of x and y in the word u, while $\ell(u)$ denotes the length of the longest prefix of u that does not contain y.

Therefore, the greatest weak forward presimulation between $\mathcal{A}$ and $\mathcal{B}$ is computed as

$$\begin{aligned} U &= \bigwedge_{u \in X^*} \tau_u^A \backslash \tau_u^B = \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & -3 & 0 \end{bmatrix} \wedge \left(\bigwedge_{h(u) = x} \begin{bmatrix} 0 & 0 & -2\ell(u) + 3 \\ 2\ell(u) - 3 & 2\ell(u) - 3 & 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & -3 & 0 \end{bmatrix} \wedge \begin{bmatrix} 0 & 0 & -\infty \\ -1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\infty \\ -1 & -3 & 0 \end{bmatrix}, \end{aligned}$$

and since

$$\sigma^B \otimes U^\top = \begin{bmatrix} 1 & -\infty & -\infty \end{bmatrix} \otimes \begin{bmatrix} 0 & -1 \\ 0 & -3 \\ -\infty & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} > \begin{bmatrix} 1 & -\infty \end{bmatrix} = \sigma^A,$$

we conclude that U is the greatest weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$. Let us note that it coincides with the greatest forward simulation between these two automata, which is computed in Example 4.2 of [9].

As already mentioned in the introduction, there are situations in which a weak simulation or bisimulation of a given type exists, while the corresponding standard simulation or bisimulation does not, whereby the converse is impossible, in accordance with Theorem 4.1. One such situation is illustrated in the following example. This fact highlights the importance of weak simulations and bisimulations, as it shows that they offer a strictly wider range of applicability in confirming containment or equivalence than their standard counterparts.

Example 5.4. *For the one-element input alphabet $X = \{x\}$, let $\mathcal{A} = (3, \sigma^A, \{M_x^A\}, \tau^A)$ and $\mathcal{B} = (2, \sigma^B, \{M_x^B\}, \tau^B)$ be max-plus automata over X given by:*

$$M_x^A = \begin{bmatrix} 3 & 5 & -\infty \\ 8 & 6 & -\infty \\ 4 & 2 & 0 \end{bmatrix}, \qquad \sigma^A = \begin{bmatrix} 5 & 4 & -\infty \end{bmatrix}, \qquad \tau^A = \begin{bmatrix} -\infty \\ -\infty \\ 1 \end{bmatrix},$$

$$M_x^B = \begin{bmatrix} 2 & -\infty \\ -\infty & -\infty \end{bmatrix}, \qquad \sigma^B = \begin{bmatrix} -\infty & 1 \end{bmatrix}, \qquad \tau^B = \begin{bmatrix} -\infty \\ 1 \end{bmatrix}.$$

The procedure for computing the pairs of τ-vectors, the corresponding residuals, as well as the infima, which ultimately yield the greatest weak forward presimulation, carried out according to Procedure 1, is presented in Table 5.

Table 5: Details of the computation of the greatest weak forward simulation in Example 5.4.

Step number	**Word length**	**Pair of τ-vectors that has been computed**	**The residual of the computed pair**	**The infimum after the current step**
1	0	$(\tau_\varepsilon^A, \tau_\varepsilon^B) = \left(\begin{bmatrix} -\infty \\ -\infty \\ 1 \end{bmatrix}, \begin{bmatrix} -\infty \\ 1 \end{bmatrix}\right)$	$\tau_\varepsilon^A \between \tau_\varepsilon^B = \begin{bmatrix} +\infty & +\infty \\ +\infty & +\infty \\ -\infty & 0 \end{bmatrix}$	$U := \begin{bmatrix} +\infty & +\infty \\ +\infty & +\infty \\ -\infty & 0 \end{bmatrix}$
2	1	$(\tau_x^A, \tau_x^B) = \left(\begin{bmatrix} -\infty \\ -\infty \\ 1 \end{bmatrix}, \begin{bmatrix} -\infty \\ -\infty \end{bmatrix}\right)$	$\tau_x^A \between \tau_x^B = \begin{bmatrix} +\infty & +\infty \\ +\infty & +\infty \\ -\infty & -\infty \end{bmatrix}$	$U := \begin{bmatrix} +\infty & +\infty \\ +\infty & +\infty \\ -\infty & -\infty \end{bmatrix}$
3	2	$(\tau_{x^2}^A, \tau_{x^2}^B) = \left(\begin{bmatrix} -\infty \\ -\infty \\ 1 \end{bmatrix}, \begin{bmatrix} -\infty \\ -\infty \end{bmatrix}\right)$ ⊠		

Therefore, the greatest weak forward presimulation between $\mathcal{A}$ and $\mathcal{B}$ is

$$U = \begin{bmatrix} +\infty & +\infty \\ +\infty & +\infty \\ -\infty & -\infty \end{bmatrix},$$

and since

$$\sigma^B \otimes U^\top = \begin{bmatrix} -\infty & 1 \end{bmatrix} \otimes \begin{bmatrix} +\infty & +\infty & -\infty \\ +\infty & +\infty & -\infty \end{bmatrix} = \begin{bmatrix} +\infty & +\infty & -\infty \end{bmatrix} > \begin{bmatrix} 5 & 4 & -\infty \end{bmatrix} = \sigma^A,$$

we get that U is the greatest weak forward simulation between $\mathcal{A}$ and $\mathcal{B}$.

On the other hand, applying the procedure from Theorem 4.1 of [9], we compute the sequence of matrices $\{U_s\}_{s\in\mathbb{N}}$ whose members are

$$U_1 = \begin{bmatrix} +\infty & +\infty \\ +\infty & +\infty \\ -\infty & 0 \end{bmatrix}, \qquad U_s = \begin{bmatrix} +\infty & -\infty \\ +\infty & -\infty \\ -\infty & -\infty \end{bmatrix}, \ \textit{for } s \in \mathbb{N},\ s \geqslant 2,$$

which means that the greatest forward presimulation between $\mathcal{A}$ and $\mathcal{B}$ is

$$\widehat{U} = \begin{bmatrix} +\infty & -\infty \\ +\infty & -\infty \\ -\infty & -\infty \end{bmatrix}$$

However, we have that

$$\sigma^B \otimes (\widehat{U})^\top = \begin{bmatrix} -\infty & 1 \end{bmatrix} \otimes \begin{bmatrix} +\infty & +\infty & -\infty \\ -\infty & -\infty & -\infty \end{bmatrix} = \begin{bmatrix} -\infty & -\infty & -\infty \end{bmatrix} \ngeqslant \begin{bmatrix} 5 & 4 & -\infty \end{bmatrix} = \sigma^A,$$

so $\widehat{U}$ is not a forward simulation, and according to Theorem 4.1 [9], there is no forward simulation between $\mathcal{A}$ and $\mathcal{B}$.

References

[1] F. Baccelli, G. Cohen, G. Olsder, J. Quadrat, Synchronization and Linearity, John Wiley and Sons Ltd, Chichester, 1992.
[2] M. Bezem, R. Nieuwenhuis, E. Rodríguez-Carbonell, Exponential behaviour of the Butkovič–Zimmermann algorithm for solving two-sided linear systems in max-algebra, Discrete Appl. Math. **156** (2008), 3506–3509.
[3] R. Boukra, S. Lahaye, J.-L. Boimond, New representations for (max,+) automata with applications to the performance evaluation of discrete event systems, IFAC Pap. Online **45** (29) (2012), 116–121.
[4] P. Butkovič, Max-linear Systems: Theory and Algorithms, Springer, London, 2010.
[5] P. Butkovič, K. Zimmermann, A strongly polynomial algorithm for solving two-sided linear systems in max-algebra, Discrete Appl. Math. **154** (2006), 437–446.
[6] M. Ćirić, J. Ignjatović, N. Damljanović, M. Bašić, Bisimulations for fuzzy automata, Fuzzy Sets Syst. **186** (2012), 100–139.
[7] M. Ćirić, J. Ignjatović, I. Jančić, N. Damljanović, Computation of the greatest simulations and bisimulations between fuzzy automata, Fuzzy Sets Syst. **208** (2012), 22–42.
[8] M. Ćirić, J. Ignjatović, P. Stanimirović, Bisimulations for weighted finite automata over semirings, Research Square, 19 Dec 2022, https://doi.org/10.21203/rs.3.rs-2386298/v1.
[9] M. Ćirić, I. Micić, J. Matejić, A. Stamenković, Simulations and bisimulations for max-plus automata, Discrete Event Dyn. Syst. **34** (2024), 269–295.
[10] T. Colcombet, L. Daviaud, F. Zuleger, Size-Change Abstraction and Max-Plus Automata, in: E. Csuhaj-Varjú, M. Dietzfelbinger, Z. Ésik (eds), Mathematical Foundations of Computer Science – MFCS 2014, Lecture Notes in Comput. Sci., vol. 8634 (2014), 208–219.
[11] N. Damljanović, M. Ćirić, J. Ignjatović, Bisimulations for weighted automata over an additively idempotent semiring, Theoret. Comput. Sci. **534** (2014), 86–100.
[12] B. Daviaud, S. Lahaye, M. Lhommeau, J. Komenda, On the existence of simulations for max-plus automata, IEEE Control Syst. Lett. **8** (2024), 694–699.
[13] L. Daviaud, Containment and Equivalence of Weighted Automata: Probabilistic and max-plus cases, in: Language and Automata Theory and Applications (LATA 2020), A. Leporati, C. Martín-Vide, D. Shapira, C. Zandron, eds, Lect. Notes Comput. Sci., vol 12038, Springer, Cham, 2020, pp. 17–32.
[14] L. Daviaud, P. Guillon, G. Merlet, Comparison of max-plus automata and joint spectral radius of tropical matrices, in: K. G. Larsen, H. L. Bodlaender, J.-F. Raskin (eds.), 42nd Int. Symp. Math. Found. Comput. Sci. (MFCS 2017), LIPIcs, Vol. 83 (2017), 19:1–19:14.
[15] M. Droste, W. Kuich, Semirings and Formal Power Series, in: Handbook of Weighted Automata, M. Droste, W. Kuich, H. Vogler, Eds., Monographs in Theoretical Computer Science, An EATCS Series, Springer, Berlin-Heidelberg, 2009, pp. 3–28.
[16] P. Eklund, J. Gutiérrez García, U. Höhle, J. Kortelainen, Semigroups in complete lattices, Developments in mathematics, Springer, Cham, 2018.
[17] S. Gaubert, Performance evaluation of (max,+) automata, IEEE Trans. Autom. Control **40** (12) (1995), 2014–2025.
[18] S. Gaubert, J. Mairesse, Modeling and Analysis of Timed Petri Nets using Heaps of Pieces, IEEE Trans. Autom. Control **44** (4) (1999), 683–698.
[19] M. Gondran, M. Minoux, Graphs, Dioids and Semirings, New Models and Algorithms, Springer, New York, 2008.
[20] J. M. Howie, Fundamentals of Semigroup Theory, Clarendon Press, Oxford, UK, 1995.
[21] I. Jančić, Weak bisimulations for fuzzy automata, Fuzzy Sets Syst. **249** (2014), 49–72.
[22] Z. Jančić, I. Micić, M. Ćirić, J. R. González de Mendívil, S. Stanimirović, Approximate determinization of fuzzy automata over the product structure by means of approximate weak simulations, Fuzzy Sets Syst. **523** (2026), 109625.
[23] Z. Jančić, I. Micić, J. Ignjatović, M. Ćirić, Further improvement of determinization methods for fuzzy finite automata, Fuzzy Sets Syst. **301** (2015), 79–102.
[24] Z. Jančić, J. R. González de Mendívil, I. Micić, M. Ćirić, Approximate children fuzzy automata over the product structure, Filomat **40**:1 (2026), 309–320.
[25] J. Komenda, S. Lahaye, J.-L. Boimond, Supervisory Control of (max,+) Automata: A Behavioral Approach, Discrete Event Dyn. Syst. **19** (4) (2009), 525–549.
[26] J. Komenda, S. Lahaye, J.-L. Boimond, (Max,+)-automata with partial observations, IFAC Pap. Online **51** (7) (2018), 192–197.
[27] J. Komenda, S. Lahaye, J.-L. Boimond, T. van den Boom, Max-Plus algebra and discrete event systems, IFAC Pap. Online **50** (1) (2017), 1784–1790.
[28] J. Komenda, S. Lahaye, J.-L. Boimond, T. van den Boom, Max-plus algebra in the history of discrete event systems, Annu. Rev. Control **45** (2018), 240–249.
[29] S. Lahaye, J. Komenda, J.-L. Boimond, Modeling of timed Petri nets using deterministic (max,+) automata, IFAC Proc. Volumes **47** (2) (2014), 471–476.
[30] S. Lahaye, J. Komenda, J.-L. Boimond, Compositions of (max,+) automata, Discrete Event Dyn. Syst. **25** (2015), 323–344.
[31] S. Lahaye, A. Lai, J. Komenda, J.-L. Boimond, A contribution to the determinization of max-plus automata, Discrete Event Dyn. Syst. **30** (2020), 155–174.
[32] S. Lombardy, J. Mairesse, Max-plus automata, in: Handbook of Automata Theory, Vol 1., J. E. Pin, ed., European Mathematical Society Publishing House GmbH – EMS Press, 2021, pp. 151–188.
[33] J. Matejić, M. Ćirić, J. Ignjatović, I. Micić, Truncating and shifting weights for max-plus automata, Axioms **15** (2026), 79.
[34] I. Micić, M. Ćirić, J. Matejić, S. Stanimirović, L. A. Nguyen, Approximate weak simulations and bisimulations for fuzzy automata over the product structure, Fuzzy Sets Syst. **485** (2024), 108959.

[35] I. Micić, S. Stanimirović, J. R. González de Mendívil, M. Ćirić, Z. Jančić, Finite determinization of fuzzy automata using a parametric product-based t-norm, Fuzzy Sets Syst. **488** (2024), 108990.
[36] J. Paseka, J. Rosický, Quantales, in: Current Research in Operational Quantum Logic: Algebras, Categories, Languages, B. Coecke, D. Moore, A. Wilce, Eds., Springer, Dordrecht, 2000; pp. 245–262.
[37] K. I. Rosenthal, Quantales and Their Applications, Pitman Research Notes in Mathematics Series, Longman Higher Education, Harlow, England, 1990.
[38] A. Stamenković, M. Ćirić, D. Djurdjanović, Weakly linear systems for matrices over the max-plus quantale, Discrete Event Dyn. Syst. **32** (2022), 1–25.
[39] A. Stamenković, M. Ćirić, J. Ignjatović, Reduction of fuzzy automata by means of fuzzy quasi-orders, Inf. Sci. **275** (2014), 168–198.
[40] S. Stanimirović, M. Ćirić, J. Ignjatović, Determinization of fuzzy automata by factorizations of fuzzy states and right invariant fuzzy quasi-orders, Inf. Sci. **469** (2018), 79–100.
[41] M. Stanković, M. Ćirić, J. Ignjatović, Hennessy-Milner type theorems for fuzzy multimodal logics over Heyting algebras, J. Mult.-Valued Logic Soft Comput. **39** (2–4) (2022), 341–379.
[42] M. Stanković, M. Ćirić, J. Ignjatović, Simulations and bisimulations for fuzzy multimodal logics over Heyting algebras, Filomat **37** (3) (2023), 711–743.
[43] L. Triska, T. Moor, Behaviour equivalent max-plus automata for timed Petri nets under open-loop race-policy semantics, Discrete Event Dyn. Syst. **31** (2021), 583–607.
[44] N. Urabe, I. Hasuo, Quantitative simulations by matrices, Inf. Comput. **252** (2017), 110–137.
[45] J. Uyttendaele, I. Van Hoeck, N. Besinovic, P. Vansteenwegen, Timetable compression using max-plus automata applied to large railway networks, TOP **31** (2023), 414–439.