

Article

Not peer-reviewed version

Truncating and Shifting Weights for Max-Plus Automata

[Jelena Matejić](#), [Miroslav Ćirić](#)^{*}, [Jelena Ignjatović](#), [Ivana Micić](#)

Posted Date: 22 December 2025

doi: 10.20944/preprints202512.1981.v1

Keywords: semiring; max-plus semiring; complete max-plus semiring; max-plus automaton



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Truncating and Shifting Weights for Max-Plus Automata

Jelena Matejić , Miroslav Ćirić * , Jelena Ignjatović  and Ivana Micić 

University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18108 Niš, Serbia

* Correspondence: miroslav.ciric@pmf.edu.rs

Abstract

In this paper, for any real number λ , we transform the complete max-plus semiring \mathbb{R}_∞ into a commutative, complete, additively idempotent semiring $\mathbb{R}_\infty^\lambda$, called the lower λ -truncation of \mathbb{R}_∞ . It is obtained by removing from \mathbb{R}_∞ all real numbers smaller than λ , inheriting the addition operation, shifting the original products by $-\lambda$, and appropriately modifying the residuum operation. The purpose of lower truncations is to transfer the iterative procedures for computing the greatest presimulations and prebisimulations between max-plus automata, in cases where they cannot be completed in a finite number of iterations over \mathbb{R}_∞ , to $\mathbb{R}_\infty^\lambda$, where they could terminate in a finite number of iterations. For instance, we prove that this necessarily happens when working with max-plus automata with integer weights. We also show how presimulations and prebisimulations computed over $\mathbb{R}_\infty^\lambda$ can be transformed into presimulations and prebisimulations between the original automata over \mathbb{R}_∞ . Although they do not play a significant role from the standpoint of computing presimulations and prebisimulations, for theoretical reasons we also introduce two types of upper truncations of the complete max-plus semiring \mathbb{R}_∞ .

Keywords: semiring; max-plus semiring; complete max-plus semiring; max-plus automaton

MSC: 15A80; 16Y60; 06F07; 68Q70; 93C65

1. Introduction

Weighted finite automata constitute a natural and widely studied generalization of classical finite automata, obtained by associating weights from a given semiring with transitions, initial states, and final states. Instead of merely accepting or rejecting input words, weighted automata assign to each word a value that aggregates the weights of all runs labeled by that word, according to the operations of the underlying semiring. This framework provides a uniform and expressive model for describing quantitative behaviors, and it has found numerous applications in areas such as language and speech processing, image analysis, control theory, performance evaluation, and the modeling of discrete-event dynamical systems. The choice of the semiring plays a central role in determining both the semantics and the range of applications of weighted automata. Classical examples include probabilistic automata over the probability semiring, cost automata over the tropical semiring, and automata over Boolean or arithmetic semirings.

Within this general setting, max-plus automata arise as a particularly important and well-studied class. They are weighted finite automata defined over the max-plus semiring, in which addition is interpreted as taking the maximum and multiplication corresponds to ordinary addition. The max-plus semiring has proved to be a powerful mathematical tool for modeling and analyzing systems in which synchronization, timing, and resource sharing are fundamental. Typical application domains include manufacturing systems, transportation networks, communication protocols, scheduling problems, and discrete-event systems. In these contexts, the max-plus operations naturally capture the propagation of delays, the accumulation of processing times, and the synchronization of concurrent events, making

max-plus models both intuitive and analytically tractable. 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. From this perspective, max-plus automata describe the worst-case or longest execution time associated with a given sequence of events. This interpretation makes them especially suitable for performance analysis and verification tasks, where extremal timing behavior is of primary interest. Due to their strong algebraic structure and rich expressive power, max-plus automata form a central object of study 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.

Two fundamental decision problems associated with weighted finite automata are the *equivalence problem* and the *containment problem*. Given two weighted automata over the same semiring, the *equivalence problem* asks whether they assign the same value to every input word, whereas the *containment problem* asks whether the value assigned by one automaton is less than or equal to the value assigned by the other for all words, provided that the semiring from which the weights are taken is ordered. These problems generalize classical language equivalence and inclusion problems, and they play a central role in verification, optimization, and model comparison. In the context of max-plus automata, these problems admit a particularly intuitive interpretation – the containment problem asks whether the behavior of one system is always no slower than that of another, while equivalence asserts that both systems exhibit identical 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. For general weighted automata, decidability and complexity heavily depend on the chosen semiring. In the max-plus setting, equivalence is known to be decidable under certain restrictions, but it becomes difficult or undecidable in more general frameworks, particularly when infinite behaviors or unrestricted weights are allowed. 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, and in some cases complete, proof techniques for establishing containment and equivalence. Simulations provide sufficient, but most often not necessary, conditions for containment, while bisimulations serve as sufficient, but most often not necessary, conditions for equivalence. Originally introduced in the setting of classical (Boolean) finite automata and labeled transition systems, simulations and bisimulations were defined as binary relations between states, typically represented by Boolean matrices. In this classical framework, a simulation relation ensures that every transition of one system can be matched by a corresponding transition of another system, whereas a bisimulation realizes mutual simulation.

When extending these concepts to weighted finite automata, and in particular to automata over the max-plus semiring, the classical Boolean relational approach proves insufficient for capturing quantitative aspects of behavior. Boolean matrices can express the existence of a correspondence between states, but they cannot encode quantitative discrepancies in accumulated weights, such as differences in execution times or costs. As a result, classical simulations often lead to overly coarse abstractions when applied to max-plus automata, and classical bisimulation may fail to reflect meaningful quantitative distinctions between systems. To overcome this limitation, a quantitative approach was developed in [11,31,40], in which simulations and bisimulations are defined as matrices over the max-plus semiring rather than as Boolean relations. In particular, two types of simulations and four types of bisimulations were defined in [11] as solutions of specific systems of matrix inequations, 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 inequations. However, solving these systems is a rather challenging task, which will be examined in more detail below.

The matrix inequations defining any of the aforementioned types of simulations and bisimulations can be divided into three groups. The system consisting of the inequations from the second and third groups is always solvable, and its solutions are called *presimulations* or *prebisimulations* of the given type. Moreover, this system always has a greatest solution, which we call the *greatest presimulation* or the *greatest prebisimulation* of that type. The inequations from the first group are used to test the greatest presimulation or prebisimulation. Specifically, if the greatest presimulation or prebisimulation of a given type is a solution of the system consisting of the inequations from the first group, then it is also the greatest simulation or bisimulation of that type. Otherwise, if the greatest presimulation or prebisimulation is not a solution of this system, then no simulation or bisimulation of that type exists.

While testing the greatest presimulation or prebisimulation on the inequalities from the first group is a relatively simple task, computing the greatest presimulation or prebisimulation is a significantly more difficult task and requires special attention. The procedure for computing the greatest presimulation or simulation of a given type between two max-plus automata, proposed in [11], consists of constructing a non-increasing sequence of matrices $\{U_s\}_{s \in \mathbb{N}}$ whose infimum \widehat{U} is the greatest simulation or prebisimulation being sought. A central place in constructing this sequence of matrices is held by the *residuum operation*, which is fully defined on the complete max-plus semiring \mathbb{R}_∞ and extends to matrices over this semiring. For this reason, the max-plus automata we work with are treated as automata over the complete max-plus semiring \mathbb{R}_∞ rather than over the usual max-plus semiring \mathbb{R}_{\max} , even if $+\infty$ does not appear in the transition matrices and the vectors of initial and final weights of these automata.

In cases where the sequence $\{U_s\}_{s \in \mathbb{N}}$ is finite, its infimum can be computed efficiently. Namely, in such cases we can find the smallest s for which $U_s = U_{s+1}$, the sequence then stabilizes at U_s , and U_s is the infimum of the sequence, as its smallest member. However, problems arise in cases where the sequence is infinite. In some of these cases, the infimum of the sequence could be determined theoretically, for instance by using the Principle of Mathematical Induction, but whenever the sequence of matrices $\{U_s\}_{s \in \mathbb{N}}$ is infinite, its infimum cannot be computed efficiently using standard algorithms and software. Even in cases where all real weights appearing in the transition matrices and in the vectors of initial and final weights of the automata we consider are positive numbers, the use of the residuum operation on \mathbb{R}_∞ in constructing each member of the sequence of matrices $\{U_s\}_{s \in \mathbb{N}}$ may lead to the appearance of negative entries in these matrices, and at certain positions in these matrices one may even obtain infinite descending sequences of real numbers converging to $-\infty$. To overcome this problem, we came up with the idea of truncating the semiring \mathbb{R}_∞ from below at some real number λ , by discarding all elements of \mathbb{R}_∞ that are smaller than λ , except for $-\infty$. Replacing \mathbb{R}_∞ with $\mathbb{R}_\infty^\lambda = [\lambda, +\infty) \cup \{-\infty, +\infty\}$ also requires a modification of the multiplication operation, which is very simply done by shifting the original products by $-\lambda$, which also entails a somewhat more substantial modification of the residuum operation. The modified residuum operation allows the aforementioned descending sequences of real numbers, which appear in the sequence of matrices $\{U_s\}_{s \in \mathbb{N}}$ and converge to $-\infty$, to become finite, roughly speaking, by setting all elements of these sequences that are smaller than λ to $-\infty$. Thus, the essence of truncating the semiring \mathbb{R}_∞ is that in cases where the computation of the greatest presimulation or bisimulation cannot be carried out in a finite number of steps, this computation is transferred to the new truncated semiring $\mathbb{R}_\infty^\lambda$, where it could be performed in a finite number of steps.

The main results of the paper are as follows. First, we prove that, for any real number λ , the set $\mathbb{R}_\infty^\lambda$, equipped with the addition operation inherited from \mathbb{R}_∞ and the multiplication operation obtained by shifting the products in \mathbb{R}_∞ by $-\lambda$, forms a commutative, complete, additively idempotent semiring, and we provide a characterization of the residuum operation in this semiring (Theorem 3.1). This semiring is called the *lower λ -truncation of \mathbb{R}_∞ with shifted multiplication*. Although only lower truncations are needed for the applications discussed above, for theoretical reasons we also deal with truncations of the semiring \mathbb{R}_∞ from above at some real number v . We introduce two types of such truncations: one in which the carrier set is $\mathbb{R}_\infty^v = (-\infty, v] \cup \{-\infty, +\infty\}$, with $+\infty$ included,

and another in which the carrier set is $\mathbb{R}_{\max}^v = (-\infty, v] \cup \{-\infty\}$, where $+\infty$ is not included and the role of the greatest element is taken by v . We show that both \mathbb{R}_{∞}^v and \mathbb{R}_{\max}^v , equipped with the addition operations inherited from \mathbb{R}_{∞} and the multiplication operations obtained by shifting the products in \mathbb{R}_{∞} by $-v$, form commutative, complete, additively idempotent semirings, and we provide characterizations of residuum operations in these semirings (Theorems 3.2 and 3.3). Although the addition and multiplication operations in both semirings are almost identical, the absence of the element $+\infty$ in \mathbb{R}_{\max}^v causes the residuum operations in \mathbb{R}_{\max}^v and \mathbb{R}_{∞}^v to differ significantly.

The key idea behind lower truncations is to find, for a max-plus automaton \mathcal{A} over \mathbb{R}_{∞} , a real number λ such that all entries of the basic transition matrices and the initial and final weight vectors of this automaton are contained in $\mathbb{R}_{\infty}^{\lambda}$, and to transform \mathcal{A} into an automaton \mathcal{A}^{λ} over $\mathbb{R}_{\infty}^{\lambda}$, which is called the *lower λ -shift* of \mathcal{A} . The original automaton \mathcal{A} and its lower λ -shift \mathcal{A}^{λ} have the same basic transition matrices and initial and final weight vectors, but their behaviors are different due to the differences between the multiplication operations in \mathbb{R}_{∞} and $\mathbb{R}_{\infty}^{\lambda}$. However, the behaviors of these automata are naturally related, and this relationship is expressed by the formula proven in Theorem 4.2. As a consequence, there exists a containment or equivalence relation between two automata over \mathbb{R}_{∞} if and only if such a relation exists between their lower λ -shifts (Theorem 4.3). In the context of computing the greatest presimulations and prebisimulations, for the transition from the original automata to their lower shifts to make sense, this transition should be justified by the efficiency of the computations performed over these lower shifts. Such a justification is provided by Theorem 4.4, which states that for max-plus automata with integer weights, the greatest presimulations and prebisimulations between their lower shifts can always be computed in a finite number of steps. On the other hand, Theorem 4.5 provides a method by which presimulations, prebisimulations, simulations, and bisimulations between the lower shifts can be converted into presimulations, prebisimulations, simulations, and bisimulations between the original automata.

We also provide numerical examples illustrating the application of lower truncations in computing the greatest presimulations and prebisimulations. These examples present a case in which the greatest presimulation over the complete max-plus semiring \mathbb{R}_{∞} cannot be computed in finitely many steps, but this becomes possible over a lower truncation of \mathbb{R}_{∞} , as well as a case in which the greatest presimulation can be computed over \mathbb{R}_{∞} in a finite number of steps, but over a lower truncation of \mathbb{R}_{∞} it can be computed in fewer steps. The examples also point to one drawback of lower truncations, namely that it may happen that a simulation exists between the original automata, but not between their lower shifts. The reason for this is that the modified residuum operation can make the greatest presimulation between the lower shifts so small that it no longer satisfies the inequalities from the first group. It is also shown that, in the context of computing the greatest presimulations and prebisimulations, the use of upper truncations does not yield results as good as those obtained by using lower truncations.

At the end of the paper, we also provide some observations on *weak simulations* and *bisimulations*. These are generalizations of ordinary simulations and bisimulations that better detect the existence of containment or equivalence between two automata, but are significantly harder to compute. Weak simulations and bisimulations are also defined by means of particular systems of matrix inequations, where these inequations can be divided into two groups. The system consisting of the inequations from the second group is always solvable, and its solutions are called *weak presimulations* or *prebisimulations* of the given type, while the greatest solution of this system, which always exists, is called the *greatest weak presimulation* or *prebisimulation*. If the greatest weak presimulation or prebisimulation is a solution of the system consisting of the inequations from the first group, then it is the greatest weak simulation or bisimulation of that type, and otherwise, if it is not a solution of that system, then no weak simulation or bisimulation of that type exists. The main problem that arises here is how to compute the greatest weak presimulation or prebisimulation, since the system that defines them may consist of infinitely many inequations, in which case they cannot be computed efficiently. The numerical examples we provide show that lower and upper truncations of the semiring \mathbb{R}_{∞} do not help in resolving this problem, and

they also point to some other aspects of the problem and suggest other possible approaches to its solution, which we will address in our future research.

It should be noted that simulations for max-plus automata first appeared in [40], where they were defined as matrices over the max-plus semiring that are solutions of systems of matrix inequations similar (though not identical) to those presented here, in Section 4. 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. [3,6]). 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 [13] 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 [16]. As we have already noted, the iterative approach to solving the systems of matrix inequations that define simulations and bisimulations, which we use here, was introduced in [11] and is based on the approach to solving somewhat more general weakly linear systems of matrix inequations developed in [36].

The paper consists of five sections. After this introductory section, Section 2 introduces the basic concepts and notation related to semirings, in particular complete additively idempotent semirings, as well as the concepts and notation concerning vectors and matrices over these semirings. In Section 3, we first recall the definitions of the max-plus semiring and the complete max-plus semiring, and then introduce the notions of lower and upper truncations of the complete max-plus semiring and prove the basic results concerning these truncations. Next, in Section 4, we first introduce the basic concepts and notation related to weighted automata over a semiring and max-plus automata, and then prove the fundamental results that enable the successful application of lower truncations in the efficient computation of presimulations and prebisimulations between max-plus automata. In the final section, Section 5, we present observations on weak simulations and bisimulations between max-plus automata, as discussed above.

2. Preliminaries

Throughout this paper, \mathbb{R} denotes the set of real numbers, \mathbb{Z} the set of integers, \mathbb{N} the set of natural numbers without zero, and \mathbb{N}_0 the set of natural numbers including zero. For $n \in \mathbb{N}$, we denote by $[1..n]$ the set of all natural numbers from 1 to n .

A *semiring* is an algebra $\mathbb{S} = (\mathbb{S}, \oplus, \otimes, 0, 1)$ with a carrier set \mathbb{S} , two binary operations \oplus and \otimes on \mathbb{S} , and two constants $0, 1 \in \mathbb{S}$ such that

- (S1) $(\mathbb{S}, \oplus, 0)$ is a commutative monoid with identity 0,
- (S2) $(\mathbb{S}, \otimes, 1)$ is a monoid with identity 1,
- (S3) $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ and $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$, for all $a, b, c \in \mathbb{S}$ (*distributivity laws*),
- (S4) $0 \otimes a = a \otimes 0 = 0$, for every $a \in \mathbb{S}$ (*absorption laws*).

We call 0 the *zero* and 1 the *identity* of the semiring \mathbb{S} . The operation \oplus is called *addition*, and the operation \otimes is called *multiplication*. If the multiplication is also commutative, then \mathbb{S} is called a *commutative semiring*. It is customary to identify an algebraic structure with its carrier set, so the carrier set of the semiring is denoted by the same symbol.

A semiring \mathbb{S} is called *additively idempotent* if its addition operation is idempotent, i.e., if $a \oplus a = a$, for every $a \in \mathbb{S}$. In such a semiring, a partial order \leq defined by

$$a \leq b \Leftrightarrow a \oplus b = b, \text{ for all } a, b \in \mathbb{S}$$

is called the *canonical partial order* on \mathbb{S} (cf. [22]). The canonical partial order is compatible both with addition and multiplication, and 0 is the least element. In other words, with respect to the canonical partial order, \mathbb{S} is a positive semiring (cf. [17,21]).

For arbitrary $a, b \in \mathbb{S}$, the sum $a \oplus b$ is the supremum of the set $\{a, b\}$, that is, the addition operation coincides with the (binary) supremum operation and (\mathbb{S}, \oplus) is an upper semilattice. In the case when (\mathbb{S}, \oplus) is a *complete upper semilattice*, that is, when every subset $\{a_i\}_{i \in I} \subseteq \mathbb{S}$ has a supremum, that supremum will be denoted by $\bigoplus_{i \in I} a_i$ and will be treated as an *infinite sum*. Note that it satisfies all the conditions of the definition of an infinite sum from [17] (that definition differs from the definition of an infinite sum given in [21, Ch. 22], which also includes infinite distributivity, while in [17] and here, infinite distributivity is considered separately).

If \mathbb{S} is an additively idempotent semiring such that (\mathbb{S}, \oplus) is a complete upper semilattice (with suprema represented as sums), and if it satisfies the *infinite distributive laws*

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

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*. This definition is consistent with the definition of a complete dioid from [2, Def. 4.32], as well as with definitions of a complete semiring from [17,21]. As we have already mentioned, complete additively idempotent semirings also possess the structure of a complete lattice, and in terms of lattice theory, structures with properties identical to those defining complete additively idempotent semirings are known as *unital quantales* (cf. [18,23,34,35]).

For a complete additively idempotent semiring \mathbb{S} and arbitrary elements $a, b \in \mathbb{S}$, the inequation $a \otimes x \leq b$, where x is an unknown taking values in \mathbb{S} , has the greatest solution, which is denoted by $a \setminus b$ and called the *left residual* of b by a . Similarly, the inequation $x \otimes a \leq b$ has the greatest solution, which is denoted by b / a and called the *right residual* of b by a . The operations \setminus and $/$ are called *residuum operations* adjoined to the multiplication \otimes . It immediately follows from those definitions that

$$a \otimes b \leq c \Leftrightarrow b \leq a \setminus c \Leftrightarrow a \leq c / b, \quad (2)$$

for all $a, b, c \in \mathbb{S}$. This formula is called the *residuation property* or the *adjunction property*.

Another important property of residuals in a complete additively idempotent semiring \mathbb{S} is the following:

$$0 \setminus a = a / 0 = \top, \quad (3)$$

for every $a \in \mathbb{S}$, where \top denotes the greatest element of \mathbb{S} .

A complete additively idempotent semiring \mathbb{S} whose multiplication operation \otimes is commutative is called a *commutative complete additively idempotent semiring*. In such a semiring we have that $a \setminus b = b / a$, for all $a, b \in \mathbb{S}$, which means that there is only one residuum operation denoted by \rightarrow , that is, $a \rightarrow b = a \setminus b = b / a$, for all $a, b \in \mathbb{S}$. In this case the residuation property becomes

$$a \otimes b \leq c \Leftrightarrow b \leq a \rightarrow c \quad (4)$$

for all $a, b, c \in \mathbb{S}$, and \rightarrow is said to be the *residuum operation adjoined to \otimes* .

Let $\mathbb{S} = (\mathbb{S}, \oplus, \otimes, 0, 1)$ be a semiring. Throughout the paper, for $m, n \in \mathbb{N}$, the set of all $m \times n$ matrices with entries in \mathbb{S} is denoted by $\mathbb{S}^{m \times n}$, and the set of all vectors of size n with entries in \mathbb{S} is denoted by \mathbb{S}^n . When vectors from \mathbb{S}^n are written as *row vectors* ($1 \times n$ matrices), then instead of \mathbb{S}^n we write $\mathbb{S}^{1 \times n}$, and when they are written as *column vectors* ($n \times 1$ matrices), then instead of \mathbb{S}^n we

write $\mathbb{S}^{n \times 1}$. For a matrix $M \in \mathbb{S}^{m \times n}$, the entry of M located in the i -th row and j -th column is denoted by $M(i, j)$, and for a vector $\alpha \in \mathbb{S}^n$, the i -th entry of α is denoted by $\alpha(i)$. The *transpose* of a matrix $M \in \mathbb{S}^{m \times n}$ is the unique matrix $M^T \in \mathbb{S}^{n \times m}$ such that $M^T(j, i) = M(i, j)$, for all $j \in [1..n]$ and $i \in [1..m]$. A matrix of an arbitrary type whose all entries are equal to the zero of \mathbb{S} is called a *zero matrix*, and an $n \times n$ matrix I_n whose entries on the main diagonal are equal to the identity of \mathbb{S} , while all the others are equal to the zero, is called the *identity matrix* of order n .

For matrices $M \in \mathbb{S}^{m \times n}$ and $N \in \mathbb{S}^{n \times p}$, their *matrix product* is a matrix $M \otimes N \in \mathbb{S}^{m \times p}$ defined, for arbitrary $i \in [1..n]$ and $k \in [1..p]$, by

$$(M \otimes N)(i, k) = \bigoplus_{j=1}^n (M(i, j) \otimes M(j, k)). \quad (5)$$

For a matrix $M \in \mathbb{S}^{m \times n}$, a row vector $\alpha \in \mathbb{S}^{1 \times m}$ and a column vector $\beta \in \mathbb{S}^{n \times 1}$, the product $\alpha \otimes M \in \mathbb{S}^{1 \times n}$ is called the *vector-matrix product* of α and M , while the product $M \otimes \beta \in \mathbb{S}^{m \times 1}$ is called the *matrix-vector product* of M and β . More precisely,

$$(\alpha \otimes M)(j) = \bigoplus_{k=1}^m (\alpha(k) \otimes M(k, j)), \quad (M \otimes \beta)(i) = \bigoplus_{k=1}^n (M(i, k) \otimes \beta(k)), \quad (6)$$

for all $j \in [1..n]$ and $i \in [1..m]$. In particular, the product $\alpha \otimes \beta$ of a row vector $\alpha \in \mathbb{S}^{1 \times m}$ and a column vector $\beta \in \mathbb{S}^{n \times 1}$ is a 1×1 matrix, which is usually identified with its single entry. In other words, we assume that $\alpha \otimes \beta$ is an element of \mathbb{S} defined by

$$\alpha \otimes \beta = \bigoplus_{i=1}^n (\alpha(i) \otimes \beta(i)). \quad (7)$$

This product is known as the *scalar product* or *dot product* of vectors α and β .

The operation of *scalar multiplication of a matrix* is defined, for $s \in \mathbb{S}$ and $M \in \mathbb{S}^{m \times n}$, in the following way: the scalar multiplication of M by s is a matrix $s \otimes M \in \mathbb{S}^{m \times n}$ given by

$$(s \otimes M)(i, j) = s \otimes M(i, j), \quad (8)$$

for all $i \in [1..m]$ and $j \in [1..n]$. The properties of scalar multiplication that we need later are

$$s \otimes (t \otimes M) = (s \otimes t) \otimes M, \quad s \otimes (M \otimes N) = (s \otimes M) \otimes N, \quad (9)$$

for all $s, t \in \mathbb{S}$, $M \in \mathbb{S}^{m \times n}$ and $N \in \mathbb{S}^{n \times p}$.

In the case when \mathbb{S} is an ordered semiring, for matrices of the same type a *matrix ordering* can be defined. It is defined coordinatewise, as follows

$$M \leq N \Leftrightarrow M(i, j) \leq N(i, j), \quad (10)$$

for all $i \in [1..m]$ and $j \in [1..n]$. When vectors from \mathbb{S}^n are considered as matrices (either row or column vectors), this definition gives the coordinatewise *ordering of vectors*.

In what follows, let $\mathbb{S} = (\mathbb{S}, \oplus, \otimes, 0, 1)$ be a complete additively idempotent semiring with residuum operations \backslash and $/$ adjoined to the multiplication \otimes . The residuum operations can also be defined for matrices and vectors with entries in \mathbb{S} , as follows. For matrices $M \in \mathbb{S}^{m \times n}$ and $P \in \mathbb{S}^{m \times p}$, the *left residual* of P by M is a matrix $M \backslash P \in \mathbb{S}^{n \times p}$ defined by

$$(M \backslash P)(j, k) = \bigwedge_{i=1}^m M(i, j) \backslash P(i, k), \quad (11)$$

for all $j \in [1..n]$ and $k \in [1..p]$. Similarly, for matrices $N \in \mathbb{S}^{n \times p}$ and $P \in \mathbb{S}^{m \times p}$, the *right residual* of P by N is a matrix $P/N \in \mathbb{S}^{m \times n}$ defined by

$$(P/N)(i, j) = \bigwedge_{k=1}^p P(i, k)/N(j, k), \quad (12)$$

for all $i \in [1..m]$ and $j \in [1..n]$. The matrix residuum operations defined in this way satisfy the following *residuation property (adjunction property)*:

$$M \otimes N \leq P \Leftrightarrow N \leq M \setminus P \Leftrightarrow M \leq P/N, \quad (13)$$

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

On the other hand, for vectors $\alpha \in \mathbb{S}^m$ and $\beta \in \mathbb{S}^n$, the *left residual* $\alpha \setminus \beta$ of β by α and the *right residual* β/α of β by α are matrices $\alpha \setminus \beta \in \mathbb{S}^{m \times n}$ and $\beta/\alpha \in \mathbb{S}^{n \times m}$ defined by

$$(\alpha \setminus \beta)(i, j) = \alpha(i) \setminus \beta(j), \quad (\beta/\alpha)(j, i) = \beta(j)/\alpha(i), \quad (14)$$

for all $i \in [1..m]$ and $j \in [1..n]$. If the multiplication \otimes is commutative, then $\beta/\alpha = (\alpha \setminus \beta)^\top$, that is, $\alpha \setminus \beta = (\beta/\alpha)^\top$. The *residuation property (adjunction property)* for vector residuum operations is

$$\alpha \otimes U \leq \beta \Leftrightarrow U \leq \alpha \setminus \beta, \quad V \otimes \alpha \leq \beta \Leftrightarrow V \leq \beta/\alpha, \quad (15)$$

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

3. Truncation and Shifting of Multiplication in the Complete Max-Plus Semiring

In the sequel, \mathbb{R} denotes the set of real numbers, while

$$\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\} \quad \text{and} \quad \mathbb{R}_{\infty} = \mathbb{R} \cup \{-\infty, +\infty\},$$

where $-\infty, +\infty \notin \mathbb{R}$ are different symbols which are interpreted as minus and plus infinity. In addition, we will also use the following notation:

$$\mathbb{R}^+ = \{a \in \mathbb{R} \mid 0 \leq a\} \quad \text{and} \quad \mathbb{R}^- = \{a \in \mathbb{R} \mid a < 0\},$$

where \leq is the usual linear ordering of real numbers and $<$ is its strict version. By the same symbol \leq we will also denote a linear ordering on \mathbb{R}_{∞} that is an extension of the usual linear ordering of real numbers such that $-\infty$ is the smallest and $+\infty$ is the greatest element. It is well-known that \mathbb{R}_{∞} , with respect to this ordering, is a complete lattice. Let \wedge and \vee be the common notations for the lattice-theoretical finite meet and join operations, and let big symbols \bigwedge and \bigvee be used to denote infinite meets and joins.

In addition, a binary operation \oplus on \mathbb{R}_{∞} , called *addition*, is defined with $\oplus = \max = \vee$, while another binary operation \otimes on \mathbb{R}_{∞} , called *multiplication*, is given by the following Cayley-like table

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

The restrictions of the operations \oplus and \otimes on \mathbb{R}_{\max} are denoted by the same symbols.

It is well-known that $\mathbb{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, -\infty, 0)$ is a commutative semiring with the zero $-\infty$ and the identity 0. This semiring is also additively idempotent, and the above linear ordering coincides with the canonical partial order. We call \mathbb{R}_{\max} the *max-plus semiring*.

On the other hand, $\mathbb{R}_{\infty} = (\mathbb{R}_{\infty}, \oplus, \otimes, -\infty, 0)$ is a commutative complete additively idempotent semiring, and it will be called the *complete max-plus semiring*. We can also consider \mathbb{R}_{∞} from the

lattice-theoretical point of view, and then, \mathbb{R}_∞ is a unital quantale with the identity 0, so sometimes it is also called the *max-plus quantale*.

As we said in the previous section, every commutative complete additively idempotent semiring has a unique residuum operation adjoined to the multiplication, and in the case of the semiring \mathbb{R}_∞ , the residuum operation \rightarrow adjoined to \otimes is given by the following Cayley-like table

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

Note that \oplus and \vee represent the same operation on \mathbb{R}_{\max} and \mathbb{R}_∞ , but we use the notation \oplus rather than \vee in order to emphasize its role in the semirings \mathbb{R}_{\max} and \mathbb{R}_∞ .

The sets $\mathbb{Z}_\infty = \mathbb{Z} \cup \{-\infty, +\infty\}$ and $\mathbb{Z}_{\max} = \mathbb{Z} \cup \{-\infty\}$ are subsets of \mathbb{R}_∞ closed under operations \oplus and \otimes , whereas \mathbb{Z}_∞ is also closed under infinite sums and the operation \rightarrow . Therefore, with the operations inherited from \mathbb{R}_∞ , which are denoted by the same symbols as the corresponding operations in \mathbb{R}_∞ , $\mathbb{Z}_\infty = (\mathbb{Z}_\infty, \oplus, \otimes, -\infty, 0)$ constitutes a commutative complete additively idempotent semiring, which we call the *complete max-plus semiring of integers*, while $\mathbb{Z}_{\max} = (\mathbb{Z}_{\max}, \oplus, \otimes, -\infty, 0)$ constitutes an additively idempotent semiring, called the *max-plus semiring of integers*.

Now we turn to the introduction of some new concepts and the presentation of the results related to them. For an arbitrary $\lambda \in \mathbb{R}$, let

$$\mathbb{R}_\infty^\lambda = [\lambda, +\infty) \cup \{-\infty, +\infty\} \quad \text{and} \quad \mathbb{R}_{\max}^\lambda = [\lambda, +\infty) \cup \{-\infty\},$$

and let the restrictions of the ordering on \mathbb{R}_∞ to $\mathbb{R}_\infty^\lambda$ and $\mathbb{R}_{\max}^\lambda$ be denoted by the same symbol \leq as the original ordering on \mathbb{R}_∞ .

Theorem 3.1. For any $\lambda \in \mathbb{R}$, let \oplus_λ be the addition operation on $\mathbb{R}_\infty^\lambda$ inherited from the addition operation on \mathbb{R}_∞ , and let \otimes_λ be the multiplication operation on $\mathbb{R}_\infty^\lambda$ defined as follows:

\otimes_λ	$b \in [\lambda, +\infty)$	$-\infty$	$+\infty$
$a \in [\lambda, +\infty)$	$a + b - \lambda$	$-\infty$	$+\infty$
$-\infty$	$-\infty$	$-\infty$	$-\infty$
$+\infty$	$+\infty$	$-\infty$	$+\infty$

Then $(\mathbb{R}_\infty^\lambda, \oplus_\lambda, \otimes_\lambda, -\infty, \lambda)$ is a commutative complete additively idempotent semiring with the zero $-\infty$ and the identity λ , and the residuum operation \rightarrow_λ on $\mathbb{R}_\infty^\lambda$ adjoined to \otimes_λ is defined by

\rightarrow_λ	$b \in [\lambda, +\infty)$		$-\infty$	$+\infty$
	$a \leq b$	$a > b$	$-\infty$	$+\infty$
$a \in [\lambda, +\infty)$	$b - a + \lambda$	$-\infty$	$-\infty$	$+\infty$
$-\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
$+\infty$	$-\infty$	$-\infty$	$-\infty$	$+\infty$

Proof. For arbitrary $a, b \in [\lambda, +\infty)$, from $a \geq \lambda$ and $b \geq \lambda$ it follows that $a + b \geq 2\lambda$, which implies that $a + b - \lambda \geq \lambda$, and this means that \otimes_λ is a well-defined operation on $\mathbb{R}_\infty^\lambda$. Let us note that $a \otimes_\lambda b = (a \otimes b) - \lambda = (a - \lambda) \otimes b = a \otimes (b - \lambda)$, for all $a, b \in [\lambda, +\infty)$. It is also clear that $\mathbb{R}_\infty^\lambda$ is closed under finite and infinite sums (where infinite sums in $\mathbb{R}_\infty^\lambda$ are also inherited from \mathbb{R}_∞), and when working with infinite sums, we will use the same notation as for infinite sums in \mathbb{R}_∞ .

It is clear that the multiplication operation \otimes_λ is commutative, and it is easy to check that it is also associative and that λ is the identity for \otimes_λ . On the other hand, since the addition operation \oplus_λ is inherited from \mathbb{R}_∞ , it is also commutative and associative, and $-\infty$ is the zero. For the sake of simplicity, in the sequel we will denote the infinite sums in $\mathbb{R}_\infty^\lambda$ and \mathbb{R}_∞ with the same symbol (without writing

the subscript λ for the sums in $\mathbb{R}_\infty^\lambda$, which does not lead to confusion because these sums coincide when applied to the elements of $\mathbb{R}_\infty^\lambda$.

To prove infinite distributivity, we first consider any $a \in [\lambda, +\infty)$ and $\{b_i\}_{i \in I} \subseteq [\lambda, +\infty)$. Then we have that

$$a \otimes_\lambda \left(\bigoplus_{i \in I} b_i \right) = (a - \lambda) \otimes \left(\bigoplus_{i \in I} b_i \right) = \bigoplus_{i \in I} ((a - \lambda) \otimes b_i) = \bigoplus_{i \in I} (a \otimes_\lambda b_i).$$

Next, if $a = -\infty$, then

$$a \otimes_\lambda \left(\bigoplus_{i \in I} b_i \right) = -\infty = \bigoplus_{i \in I} (a \otimes_\lambda b_i),$$

for each $\{b_i\}_{i \in I} \subseteq \mathbb{R}_\infty^\lambda$.

Further, if $a \in \mathbb{R}_\infty^\lambda$ such that $a \neq -\infty$, and if $\{b_i\}_{i \in I} \subseteq \mathbb{R}_\infty^\lambda$ such that $b_j = +\infty$, for some $j \in I$, then $a \otimes_\lambda b_j = +\infty$ and

$$a \otimes_\lambda \left(\bigoplus_{i \in I} b_i \right) = a \otimes_\lambda +\infty = +\infty = \bigoplus_{i \in I} (a \otimes_\lambda b_i).$$

Finally, if $a \in \mathbb{R}_\infty^\lambda$ such that $a \neq -\infty$, and $\{b_i\}_{i \in I} \subseteq [\lambda, +\infty) \cup \{-\infty\}$, then

$$a \otimes_\lambda \left(\bigoplus_{i \in I} b_i \right) = a \otimes_\lambda \left(\bigoplus_{j \in J} b_j \right) = \bigoplus_{j \in J} (a \otimes_\lambda b_j) = \bigoplus_{i \in I} (a \otimes_\lambda b_i).$$

where $J = \{j \in I \mid b_j \neq -\infty\}$. Here we used the fact that infinite sums we work with are suprema, and all terms in a supremum which are equal to the smallest element (here $-\infty$) can be omitted, while in the case when the set J is empty, we used the usual convention that the supremum of the empty set is equal to the smallest element.

Therefore, we have completed the proof of the statement that $\mathbb{R}_\infty^\lambda$ is a commutative complete additively idempotent semiring with the zero $-\infty$ and the identity λ .

It remains to prove that \rightarrow_λ is the residuum operation on $\mathbb{R}_\infty^\lambda$ adjoined to the multiplication operation \otimes_λ . This will be done by proving that $a \rightarrow_\lambda b$ is the greatest solution of the inequation $a \otimes_\lambda x \leq b$, for arbitrary $a, b \in \mathbb{R}_\infty^\lambda$. Before we prove it, it should be noted that for $a, b \in [\lambda, +\infty)$ such that $b \geq a$ we have that $b - a \geq 0$, so $b - a + \lambda \geq \lambda$, and this means that \rightarrow_λ is a well-defined operation on $\mathbb{R}_\infty^\lambda$.

In the case when $a, b \in [\lambda, +\infty)$ such that $b \geq a$ we have that

$$a \otimes_\lambda x \leq b \Leftrightarrow a + x - \lambda \leq b \Leftrightarrow x \leq b - a + \lambda, \quad (16)$$

and hence, $b - a + \lambda = a \rightarrow_\lambda b$ is the greatest solution of $a \otimes_\lambda x \leq b$. On the other hand, in the case when $a, b \in [\lambda, +\infty)$ such that $b < a$ we have that $b - a + \lambda < \lambda$, which means that $b - a + \lambda \notin [\lambda, +\infty) \cup \{+\infty\}$, and from (16) it follows that $-\infty = a \rightarrow_\lambda b$ is the only solution of $a \otimes_\lambda x \leq b$. Therefore, it is the greatest solution of that inequation.

If $a \in [\lambda, +\infty)$ and $b = -\infty$, then in a similar way we conclude that $-\infty = a \rightarrow_\lambda b$ is the only solution of $a \otimes_\lambda x \leq b$, so it is the greatest solution of that inequation.

Further, if $a = -\infty$ then $a \otimes_\lambda x = -\infty \leq b$, for all $x, b \in \mathbb{R}_\infty^\lambda$, so each $x \in \mathbb{R}_\infty^\lambda$ is a solution of $a \otimes_\lambda x \leq b$, and this means that the greatest solution of that inequation is $+\infty = a \rightarrow_\lambda b$.

On the other hand, if $b = +\infty$, then for every $a \in \mathbb{R}_\infty^\lambda$ we have that any $x \in \mathbb{R}_\infty^\lambda$ is a solution of $a \otimes_\lambda x \leq b$, so the greatest solution of that inequation is $+\infty = a \rightarrow_\lambda b$.

Finally, in the case when $a = +\infty$ and $b \in \mathbb{R}_\infty^\lambda$ such that $b \neq +\infty$, we have that

$$a \otimes_\lambda x = \begin{cases} +\infty & \text{if } x \in \mathbb{R}_\infty^\lambda, x \neq -\infty, \\ -\infty & \text{if } x = -\infty, \end{cases}$$

whence it follows that $-\infty = a \rightarrow_\lambda b$ is the only solution of $a \otimes_\lambda x \leq b$, so it is the greatest solution of that inequation.

Having exhausted all possible cases, we conclude that \rightarrow_λ is the residuum operation on $\mathbb{R}_\infty^\lambda$ adjoined to \otimes_λ . \square

We will call the semiring $\mathbb{R}_\infty^\lambda$ the *lower truncation* of the semiring \mathbb{R}_∞ by means of λ , with *shifted multiplication*, or the *lower λ -truncation* for short. We will also refer to $\mathbb{R}_\infty^\lambda$ as a *lower truncated complete max-plus semiring*. If $\lambda \in \mathbb{Z}$, then $\mathbb{Z}_\infty^\lambda = \{z \in \mathbb{Z} \mid \lambda \leq z\} \cup \{-\infty, +\infty\}$ is a subset of $\mathbb{R}_\infty^\lambda$ closed under the operations \oplus_λ , \otimes_λ and \rightarrow_λ , as well as under infinite sums, so $\mathbb{Z}_\infty^\lambda = (\mathbb{Z}_\infty^\lambda, \oplus_\lambda, \otimes_\lambda, -\infty, \lambda)$ constitutes a commutative complete additively idempotent semiring, which will be called the *lower λ -truncation of \mathbb{Z}_∞ with shifted multiplication*.

An important special case of lower λ -truncations of \mathbb{R}_∞ is the one where $\lambda = 0$. The semiring \mathbb{R}_∞^0 has already been studied in [11], where it was denoted by \mathbb{R}_∞^+ . It is clear that the multiplication \otimes_0 in \mathbb{R}_∞^+ is a restriction of the multiplication \otimes on \mathbb{R}_∞ , but the residuum operation \rightarrow^0 on \mathbb{R}_∞^+ differs from the residuum operation \rightarrow on \mathbb{R}_∞ and is given by the following table:

	$b \in [0, +\infty)$			
\rightarrow_0	$a \leq b$	$a > b$	$-\infty$	$+\infty$
$a \in [0, +\infty)$	$b - a$	$-\infty$	$-\infty$	$+\infty$
$-\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
$+\infty$	$-\infty$	$-\infty$	$-\infty$	$+\infty$

In the sequel, we make few remarks regarding lower truncated semirings $\mathbb{R}_\infty^\lambda$.

Remark 3.1. In the case when $\lambda < 0$, the interval $[\lambda, +\infty)$ is not closed under the operation \otimes . Namely, for negative $a, b \in [\lambda, +\infty)$ it may happen that $a \otimes b = a + b < \lambda$, which is why we had to shift $a \otimes b = a + b$ by $-\lambda$ and bring it back into $[\lambda, +\infty)$.

In the case when $\lambda \geq 0$, the interval $[\lambda, +\infty)$ is closed under the operation \otimes , and it may seem that one could retain the original operation \otimes on $[\lambda, +\infty)$, with no need to shift this operation by $-\lambda$. However, the problem is that in that case we would lose the identity. Therefore, even when $\lambda \geq 0$, for $a, b \in [\lambda, +\infty)$ it is necessary to shift $a \otimes b = a + b$ by $-\lambda$, thereby ensuring that the new operation has an identity. It is clear that the operation \otimes is retained on $[\lambda, +\infty)$ only in the case when $\lambda = 0$.

It should also be noted that, although the interval $[\lambda, +\infty)$ has the smallest element λ , we must retain $-\infty$ as the smallest element, that is, as the zero in a semiring, because without $-\infty$ we would not obtain a semiring. Namely, if $-\infty$ were omitted, then λ would simultaneously serve as both the zero and the identity, which is possible for a semiring only in the case of a single-element semiring.

Remark 3.2. The set $\mathbb{R}_{\max}^\lambda$, equipped with the addition and multiplication operations inherited from $\mathbb{R}_\infty^\lambda$, is a commutative additively idempotent semiring with zero $-\infty$ and identity λ , but this semiring is not a complete additively idempotent semiring because it does not have the greatest element. Consequently, the residuum operation is not defined for all pairs of elements from $\mathbb{R}_{\max}^\lambda$, since the inequation $a \otimes_\lambda x \leq b$ does not have the greatest solution in $\mathbb{R}_{\max}^\lambda$ for all pairs of elements from $\mathbb{R}_{\max}^\lambda$. More precisely, this inequation does not have the greatest solution whenever $a = -\infty$, although in that case every element from $\mathbb{R}_{\max}^\lambda$ is a solution of this inequation.

We now turn to the case when the complete max-plus semiring is truncated from above. For an arbitrary $v \in \mathbb{R}$, let

$$\mathbb{R}_{\max}^v = (-\infty, v] \cup \{-\infty\} \quad \text{and} \quad \mathbb{R}_\infty^v = (-\infty, v] \cup \{-\infty, +\infty\},$$

and let the restrictions of the ordering on \mathbb{R}_∞ to \mathbb{R}_∞^v and \mathbb{R}_{\max}^v be denoted by the same symbol \leq as the original ordering on \mathbb{R}_∞ .

Theorem 3.2. For any $v \in \mathbb{R}$, let \oplus_v be the addition operation on \mathbb{R}_{\max}^v inherited from the addition operation on \mathbb{R}_{\max} , and let \otimes_v be an operation on \mathbb{R}_{\max}^v defined by the following Cayley-like table:

\otimes_v	$b \in (-\infty, v]$	$-\infty$
$a \in (-\infty, v]$	$a + b - v$	$-\infty$
$-\infty$	$-\infty$	$-\infty$

Then $(\mathbb{R}_{\max}^v, \oplus_v, \otimes_v, -\infty, v)$ is a commutative complete additively idempotent semiring with the zero $-\infty$ and the identity v , and the residuum operation \rightarrow_v on \mathbb{R}_{\max}^v adjoined to \otimes_v is given by

	$b \in (-\infty, v]$		
\rightarrow_v	$a \geq b$	$a < b$	$-\infty$
$a \in (-\infty, v]$	$b - a + v$	v	$-\infty$
$-\infty$	v	v	v

Proof. We can prove this theorem in a similar way as Theorem 3.1, so its proof will be omitted. \square

We call the semiring \mathbb{R}_{\max}^v the *upper truncation* of the semiring \mathbb{R}_{\max} by means of v , with *shifted multiplication*, or the *upper v -truncation* for short. We will also refer to \mathbb{R}_{\max}^v as the *upper truncated max-plus semiring*.

Remark 3.3. Unlike the semiring $\mathbb{R}_{\infty}^\lambda$, where we had to keep $-\infty$ as the smallest element, in the construction of the semiring \mathbb{R}_{\max}^v there was no risk of omitting $+\infty$ and thereby making v simultaneously the identity and the greatest element.

In the terminology of lattice theory, \mathbb{R}_{\max}^v is an *integral commutative quantale*, that is, a commutative quantale with an identity which is identical to the greatest element. Even more commonly, such an algebraic structure is called a *complete residuated lattice*.

However, there will be no problems even if we keep $+\infty$ and define the multiplication operation on \mathbb{R}_{∞}^v in a similar way as on $\mathbb{R}_{\infty}^\lambda$, which will be done in the next theorem.

Theorem 3.3. For any $v \in \mathbb{R}$, let \oplus_v be the addition operation on \mathbb{R}_{∞}^v inherited from the addition operation on \mathbb{R}_{∞} , and let \otimes_v be an operation on \mathbb{R}_{∞}^v defined by the following Cayley-like table:

\otimes_v	$b \in (-\infty, v]$	$-\infty$	$+\infty$
$a \in (-\infty, v]$	$a + b - v$	$-\infty$	$+\infty$
$-\infty$	$-\infty$	$-\infty$	$-\infty$
$+\infty$	$+\infty$	$-\infty$	$+\infty$

Then $(\mathbb{R}_{\infty}^v, \oplus_v, \otimes_v, -\infty, v)$ is a commutative complete additively idempotent semiring with the zero $-\infty$ and the identity v , and the residuum operation \rightarrow_v on \mathbb{R}_{∞}^v adjoined to \otimes_v is given by

	$b \in (-\infty, v]$			
\rightarrow_v	$a \geq b$	$a < b$	$-\infty$	$+\infty$
$a \in (-\infty, v]$	$b - a + v$	v	$-\infty$	$+\infty$
$-\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
$+\infty$	$-\infty$	$-\infty$	$-\infty$	$+\infty$

Proof. This theorem can also be proved in a similar way as Theorem 3.1, so the proof will be omitted. \square

The semiring \mathbb{R}_{∞}^v will be called the *upper truncation* of the semiring \mathbb{R}_{∞} by means of v , with *shifted multiplication*, or briefly the *upper v -truncation*. We also refer to \mathbb{R}_{∞}^v as the *upper truncated complete max-plus semiring*. In order to simplify the terminology, the semiring \mathbb{R}_{\max}^v will be treated not only as an upper v -truncation of \mathbb{R}_{\max} , but also as an upper v -truncation of \mathbb{R}_{∞} .

For an arbitrary $v \in \mathbb{Z}$, the sets

$$\mathbb{Z}_\infty^v = \{z \in \mathbb{Z} \mid z \leq v\} \cup \{-\infty, +\infty\} \quad \text{and} \quad \mathbb{Z}_{\max}^v = \{z \in \mathbb{Z} \mid z \leq v\} \cup \{-\infty\}$$

are subsets of \mathbb{R}_∞^v and \mathbb{R}_{\max}^v closed under the operations \oplus_v , \otimes_v and \rightarrow_v or \rightarrow_v , as well as under infinite sums. Thus, $\mathbb{Z}_\infty^v = (\mathbb{Z}_\infty^v, \oplus_v, \otimes_v, -\infty, v)$ and $\mathbb{Z}_{\max}^v = (\mathbb{Z}_{\max}^v, \oplus_v, \otimes_v, -\infty, v)$ constitute commutative complete additively idempotent semirings, which are respectively called the *upper v -truncations* of \mathbb{Z}_∞ and \mathbb{Z}_{\max} with shifted multiplication.

Remark 3.4. Although the multiplication operations on $\mathbb{R}_\infty^\lambda$ and \mathbb{R}_∞^v are similarly defined, and the multiplication operation on \mathbb{R}_{\max}^v is the restriction of the multiplication operation on \mathbb{R}_∞^v , Theorems 3.1, 3.2 and 3.3 show that the residuum operations on $\mathbb{R}_\infty^\lambda$, \mathbb{R}_∞^v , and \mathbb{R}_{\max}^v differ substantially.

4. Application of Truncation and Shifting to Max-Plus Automata

Let X a non-empty set, which we call an *alphabet* and whose elements are called *letters*. The set of all sequences of elements of X , 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 X^* 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 $f : X^* \rightarrow S$ is called a *word function*. If the set S is ordered, with an ordering \leq , then word functions can also be ordered pointwise, that is, for two word functions $f, g : X^* \rightarrow S$ we set that $f \leq g$ if $f(u) \leq g(u)$, for every word $u \in X^*$.

Further, let \mathbb{S} be a semiring and X an alphabet. A *weighted finite automaton* over \mathbb{S} and X is a quadruple $\mathcal{A} = (m, \sigma^A, \{M_x^A\}_{x \in X}, \tau^A)$, where

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

From the basic transition matrices, which correspond to the letters from the alphabet X , *compound transition matrices* are built, which correspond to the words from X^* . This is done in the following way. For any word $u = x_1 x_2 \dots x_k \in X^*$, where $x_1, x_2, \dots, x_k \in X$, the compound transition matrix $M_u^A \in \mathbb{S}^{m \times m}$ is defined by

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

and for the empty word ε we define M_ε^A to be the identity matrix I_m of order m .

The *behavior* of a weighted finite automaton $\mathcal{A} = (m, \sigma^A, \{M_x^A\}_{x \in X}, \tau^A)$ is a word function $\llbracket \mathcal{A} \rrbracket : X^* \rightarrow \mathbb{S}$ defined as follows: For $u = x_1 x_2 \dots x_k \in X^+$, where $x_1, x_2, \dots, x_k \in X$, we set

$$\llbracket \mathcal{A} \rrbracket(u) = \sigma^A \otimes M_u^A \otimes \tau^A = \sigma^A \otimes M_{x_1}^A \otimes M_{x_2}^A \otimes \dots \otimes M_{x_k}^A \otimes \tau^A, \quad (17)$$

and in addition,

$$\llbracket \mathcal{A} \rrbracket(\varepsilon) = \sigma^A \otimes M_\varepsilon^A \otimes \tau^A = \sigma^A \otimes \tau^A. \quad (18)$$

We also say that $\llbracket \mathcal{A} \rrbracket$ is the *word function computed by \mathcal{A}* .

For a weighted finite automaton $\mathcal{A} = (m, \sigma^A, \{M_x^A\}_{x \in X}, \tau^A)$, its *reverse automaton* $\overline{\mathcal{A}}$ is defined as a weighted finite automaton of the same dimension m , obtained from \mathcal{A} by mutually swapping the vectors σ^A and τ^A , and replacing each transition matrix M_x^A with its transpose.

One of the main general problems in computer science is the problem of comparing the behavior of two computational systems, algorithms, or models of computation, known as the *comparison problem*. That problem generally manifests itself in two main forms. One of them is the *equivalence problem*, where one must determine whether two computational entities produce the same output or exhibit the

same behavior for every possible input. The other is the *containment problem*, where it is necessary to determine whether the behavior or output of one computational entity is completely included within that of another. These problems have a broad range of applications, for example in model checking and verification (to verify whether an implementation matches its specification), compiler optimization (to ensure transformed code behaves the same as original code), and other areas of computer science. In the context of weighted finite automata, the equivalence problem requires determining whether two given automata \mathcal{A} and \mathcal{B} have the same behavior, that is, whether $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{B} \rrbracket$, while the containment problem requires determining whether $\llbracket \mathcal{A} \rrbracket \leq \llbracket \mathcal{B} \rrbracket$ (under the condition that the underlying semiring is ordered). If $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{B} \rrbracket$, we say that \mathcal{A} and \mathcal{B} are *equivalent automata*.

However, in many cases, including most types of weighted finite automata, the equivalence and containment problems are undecidable or computationally hard (cf. [14,15]). In such situations, it is natural to search for procedures for determining containment or equivalence that may not work in all cases, but can be implemented efficiently. The most known such procedures are based on the concepts of simulation and bisimulation.

Let two weighted finite automata $\mathcal{A} = (m, \sigma^A, \{M_x^A\}_{x \in X}, \tau^A)$ and $\mathcal{B} = (n, \sigma^B, \{M_x^B\}_{x \in X}, \tau^B)$ over an ordered semiring \mathbb{S} be given. A matrix $U \in \mathbb{S}^{m \times n}$ is said to be a *forward simulation* between automata \mathcal{A} and \mathcal{B} if it satisfies the following conditions:

$$\begin{aligned} \text{(fs1)} \quad & \sigma^A \leq \sigma^B \otimes U^\top, \\ \text{(fs2)} \quad & U^\top \otimes M_x^A \leq M_x^B \otimes U^\top, \text{ for all } x \in X, \\ \text{(fs3)} \quad & U^\top \otimes \tau^A \leq \tau^B. \end{aligned}$$

Conditions (fs1), (fs2), and (fs3) can also be treated as a system of matrix inequations, with an unknown matrix U taking values in $\mathbb{S}^{m \times n}$, and forward simulations are also defined as solutions of this system. The second type of simulations are *backward simulations* between \mathcal{A} and \mathcal{B} , which can be defined as forward simulations between the reverse automata of \mathcal{A} and \mathcal{B} . By combining the concepts of forward and backward simulations for the matrix U and its transpose U^\top , four types of bisimulations are defined. The full definitions of backward simulations and the four types of bisimulations, given via systems of matrix inequalities of a similar form, can be found in [10,11]. Here we will focus only on forward simulations.

Informally speaking, simulations between \mathcal{A} and \mathcal{B} allow automaton \mathcal{B} to mimic the moves of automaton \mathcal{A} , and if they exist, they testify to the presence of containment relation between \mathcal{A} and \mathcal{B} (i.e., they confirm that $\llbracket \mathcal{A} \rrbracket \leq \llbracket \mathcal{B} \rrbracket$). On the other hand, bisimulations also allow reverse mimicking, and if they exist, they detect the presence of equivalence between \mathcal{A} and \mathcal{B} . It is also worth noting that simulations and bisimulations, represented by matrices, specify the degree of connectedness between the states of automata \mathcal{A} and \mathcal{B} , which is generally not the case when containment or equivalence is determined in some other way.

The fundamental problem related to simulations and bisimulations is how to test the existence of a simulation or bisimulation of a given type, and, in cases where it exists, how to compute the greatest simulation or bisimulation of that type. In other words, it is the problem of testing the existence of a solution to the corresponding system of matrix inequations and computing the greatest solution, if it exists. Here, this problem will be considered in the context of weighted finite automata over the complete max-plus semiring \mathbb{R}_∞ , which we call *max-plus automata*. The name *max-plus automaton* will also be used for weighted finite automata over certain related semirings, such as the max-plus semiring \mathbb{R}_{\max} , the above introduced truncations $\mathbb{R}_\infty^\lambda$, \mathbb{R}_∞^v and \mathbb{R}_{\max}^v of \mathbb{R}_∞ and \mathbb{R}_{\max} , as well as semirings \mathbb{Z}_∞ and \mathbb{Z}_{\max} , and their truncations $\mathbb{Z}_\infty^\lambda$, \mathbb{Z}_∞^v and \mathbb{Z}_{\max}^v . Max-plus automata originated as a means of representing the behaviour of timed discrete event systems with synchronization of tasks and resource sharing, such as, for instance, production systems, railroad networks, urban traffic networks, queuing systems, array processors, and others. The motivation for using the max-plus semiring as the structure of weights also stems from the fact that many phenomena, like synchronization, which are nonlinear

in the classical systems theory, become linear by moving from the field of real numbers to the max-plus semiring. For information on the general properties of max-plus automata, as well as bibliographic notes about them, we refer to the papers [4,11,12,15,16,19,25–32,39–41].

In applications of max-plus automata within the theory of discrete event dynamic systems, letters from the input alphabet X are called *events*, the initial weights vector is called the *vector of initial delays*, the final weights vector is called the *vector of final delays*, and transition matrices are called *matrices of transition times*. In addition, word functions taking values in \mathbb{R}_∞ are called *daters* (cf. [19,26]). For a sequence of events $u \in X^+$, $\llbracket \mathcal{A} \rrbracket(u)$ can be interpreted as the date when that sequence of events is completed, and $\llbracket \mathcal{A} \rrbracket(u) = +\infty$ means that the sequence of events will never be completed.

A procedure for testing the existence of forward simulations for max-plus automata over \mathbb{R}_∞ , and for computing the greatest forward simulation when it exists, was proposed in [11]. The main part of this procedure is the construction of a non-increasing sequence of matrices $\{U_s\}_{s \in \mathbb{N}}$ from $\mathbb{R}_\infty^{m \times n}$, defined inductively as follows:

$$U_1 = \tau^A \setminus \tau^B = (\tau^B / \tau^A)^\top, \quad U_{s+1} = U_s \wedge \left(\bigwedge_{x \in X} ((M_x^B \otimes U_s^\top) / M_x^A)^\top \right). \quad (19)$$

The infimum of that sequence, denoted by \hat{U} , is the greatest solution of the system of inequations consisting of the inequations from (fs2) and (fs3). Solutions of this system are called *forward presimulations* between \mathcal{A} and \mathcal{B} (cf. [37,38]), and therefore, \hat{U} is the greatest forward presimulation.

It is interesting to note that U_1 is the greatest solution of the linear inequation (fs3), and when U_s is given, for some $s \in \mathbb{S}$, then U_{s+1} is computed as the infimum of U_s and the greatest solution of the system of linear inequations

$$U^\top \otimes M_x^A \leq M_x^B \otimes U_s^\top, \quad \text{for all } x \in X,$$

obtained from (fs2) by replacing the unknown matrix U on the right-hand sides of the inequalities with the matrix U_s .

If the greatest forward presimulation \hat{U} is a non-zero matrix (it contains an entry different from $-\infty$) and satisfies condition (fs1), then \hat{U} is the greatest forward simulation between \mathcal{A} and \mathcal{B} . Otherwise, if \hat{U} is either the zero matrix (all its entries are $-\infty$) or does not satisfy (fs1), then there exists no forward simulation between \mathcal{A} and \mathcal{B} .

Obviously, the key problem that arises in the practical application of this procedure is the efficient computation of the greatest forward presimulation \hat{U} . One way to efficiently compute \hat{U} can be applied in situations where $s \in \mathbb{N}$ can be found such that $U_s = U_{s+1}$. If this happens, then one obtains that $\hat{U} = U_s$. Therefore, after constructing the matrix U_{s+1} from U_s , we need to check whether they are equal, and once we find the smallest index s such that $U_{s+1} = U_s$, if such an index exists, we conclude that $\hat{U} = U_s$ and the procedure for constructing the sequence (19) terminates. However, it may happen that all members of the sequence are different and that the sequence is infinite. For instance, for some $i \in [1..m]$ and $j \in [1..n]$ it may happen that $\{U_s(i, j)\}_{s \in \mathbb{N}}$ is an infinite strictly decreasing sequence of real numbers that converges to $-\infty$. Behind our idea of lower λ -truncation lies the effort to make such sequences finite, to stop them whenever their members fall below λ . We will see that such truncation works particularly well when dealing with max-plus automata whose vectors of initial and final delays and matrices of transition times have integer entries, that is, when working with max-plus automata over \mathbb{Z}_∞ .

Before we move on to max-plus automata over truncated max-plus semirings, we will show how matrices over these semirings are multiplied.

For arbitrary $r \in \mathbb{R}$ and $k \in \mathbb{N}_0$, in order to distinguish between the k -th multiplicative power of r in the semiring \mathbb{R}_∞ and the k -th multiplicative power of r in the field of real numbers, the k -th multiplicative power of r in the semiring \mathbb{R}_∞ will be denoted by $r^{[k]}$.

Theorem 4.1. For an arbitrary $\xi \in \mathbb{R}$ let \mathbb{R}_∞^ξ be either a lower or upper ξ -truncation of \mathbb{R}_∞ , and for an arbitrary $s \in \mathbb{N}$ let M_1, M_2, \dots, M_s be matrices with entries in \mathbb{R}_∞^ξ such that the matrix product $M_1 \otimes_\xi M_2 \otimes_\xi \dots \otimes_\xi M_s$ exists. Then

$$M_1 \otimes_\xi M_2 \otimes_\xi \dots \otimes_\xi M_s = (-\xi)^{[s-1]} \otimes (M_1 \otimes M_2 \otimes \dots \otimes M_s), \quad (20)$$

where $(-\xi)^{[s-1]}$ denotes the $(s-1)$ st power of $-\xi$ in the multiplicative monoid of \mathbb{R}_∞^ξ .

Proof. Before proving the theorem, we will show that $a \otimes_\xi b = (-\xi) \otimes a \otimes b$, for all $a, b \in \mathbb{R}_\infty^\xi$. Indeed, for $a, b \in [\xi, +\infty)$ we have that $a \otimes_\xi b = a + b - \xi = (-\xi) + a + b = (-\xi) \otimes a \otimes b$, whereas in all other cases it follows that $a \otimes_\xi b = a \otimes b = (-\xi) \otimes a \otimes b$, since either $a \otimes b = -\infty$ or $a \otimes b = +\infty$, whereby $(-\xi) \otimes -\infty = -\infty$ and $(-\xi) \otimes +\infty = +\infty$.

We can now proceed to the proof of the theorem itself, which will be proved by induction on s . It is clear that (20) is true for $s = 1$, seeing that

$$(-\xi)^{[1-1]} \otimes M_1 = (-\xi)^{[0]} \otimes M_1 = 0 \otimes M_1 = M_1.$$

To make the proof clearer, we will also prove the case $s = 2$. Therefore, we consider the matrices $M_1 \in \mathbb{S}^{m \times n}$ and $M_2 \in \mathbb{S}^{n \times p}$. Then for arbitrary $i \in [1..m]$ and $k \in [1..p]$ we obtain

$$\begin{aligned} (M_1 \otimes_\xi M_2)(i, k) &= \bigoplus_{j=1}^n (M_1(i, j) \otimes_\xi M_2(j, k)) = \bigoplus_{j=1}^n ((-\xi) \otimes M_1(i, j) \otimes M_2(j, k)) \\ &= (-\xi) \otimes \left(\bigoplus_{j=1}^n (M_1(i, j) \otimes M_2(j, k)) \right) = (-\xi) \otimes ((M_1 \otimes M_2)(i, k)) \\ &= ((-\xi) \otimes M_1 \otimes M_2)(i, k), \end{aligned}$$

and hence, $M_1 \otimes_\xi M_2 = (-\xi) \otimes M_1 \otimes M_2$, which is what was to be proved.

Assume now that the assertion is valid for some $s \in \mathbb{N}$, $s \geq 2$, i.e., that (20) holds. Let M_{s+1} be a matrix with entries in \mathbb{R}_∞^ξ such that the product $M_1 \otimes_\xi M_2 \otimes_\xi \dots \otimes_\xi M_s \otimes_\xi M_{s+1}$ exists. According to the induction hypothesis and the properties of scalar multiplication, we get

$$\begin{aligned} M_1 \otimes_\xi M_2 \otimes_\xi \dots \otimes_\xi M_s \otimes_\xi M_{s+1} &= (M_1 \otimes_\xi M_2 \otimes_\xi \dots \otimes_\xi M_s) \otimes_\xi M_{s+1} \\ &= (-\xi) \otimes ((M_1 \otimes_\xi M_2 \otimes_\xi \dots \otimes_\xi M_s) \otimes M_{s+1}) \\ &= (-\xi) \otimes \left(((-\xi)^{[s-1]} \otimes (M_1 \otimes M_2 \otimes \dots \otimes M_s)) \otimes M_{s+1} \right) \\ &= (-\xi) \otimes \left((-\xi)^{[s-1]} \otimes (M_1 \otimes M_2 \otimes \dots \otimes M_s \otimes M_{s+1}) \right) \\ &= \left((-\xi) \otimes (-\xi)^{[s-1]} \right) \otimes (M_1 \otimes M_2 \otimes \dots \otimes M_s \otimes M_{s+1}) \\ &= (-\xi)^{[s]} \otimes (M_1 \otimes M_2 \otimes \dots \otimes M_s \otimes M_{s+1}) \end{aligned}$$

This completes the proof of the theorem. \square

Let $\mathcal{A} = (m, \sigma^A, \{M_x^A\}_{x \in X}, \tau^A)$ be an arbitrary max-plus automaton over the complete max-plus semiring \mathbb{R}_∞ , and let $\mathbb{R}_\infty^\lambda$ be the lower λ -truncation of \mathbb{R}_∞ , where λ is a real number such that all entries of matrices M_x^A , $x \in X$, and vectors σ^A and τ^A , belong to $\mathbb{R}_\infty^\lambda$. The max-plus automaton over $\mathbb{R}_\infty^\lambda$ having the same transition matrices M_x^A , $x \in X$, and initial and final weights vectors σ^A and τ^A as \mathcal{A} , will be denoted by \mathcal{A}^λ and called the *lower λ -shift* of \mathcal{A} .

Similarly, if \mathbb{R}_∞^v is the upper v -truncation of \mathbb{R}_∞ , where v is a real number such that all entries of matrices M_x^A , $x \in X$, and vectors σ^A and τ^A , belong to \mathbb{R}_∞^v , then the max-plus automaton over \mathbb{R}_∞^v having the same transition matrices M_x^A , $x \in X$, and initial and final weights vectors σ^A and τ^A as \mathcal{A} , will be denoted by \mathcal{A}^v and called the *upper v -shift* of \mathcal{A} . This definition can also be adapted to the case when the upper v -shift \mathcal{A}^v is regarded as an automaton over \mathbb{R}_{\max}^v .

When we say that \mathcal{A}^λ is the lower λ -shift of \mathcal{A} , this includes the assumption that \mathcal{A}^λ is well-defined, i.e., that all entries of the transition matrices and initial and final weights vectors of \mathcal{A} belong to the lower λ -truncation $\mathbb{R}_\infty^\lambda$ of \mathbb{R}_∞ . Similarly, when we say that \mathcal{A}^ν is the upper ν -shift of \mathcal{A} , this includes the assumption that \mathcal{A}^ν is well-defined, i.e., that all entries of the transition matrices and initial and final weights vectors belong to the upper ν -truncation \mathbb{R}_∞^ν of \mathbb{R}_∞ .

Theorem 4.2. *Let \mathcal{A} be a max-plus automaton over the complete max-plus semiring \mathbb{R}_∞ , and let $\xi \in \mathbb{R}$. If \mathcal{A}^ξ is either the lower or the upper ξ -shift of \mathcal{A} , then*

$$\llbracket \mathcal{A}^\xi \rrbracket(u) = (-\xi)^{[s+1]} \otimes \llbracket \mathcal{A} \rrbracket(u), \quad (21)$$

for every word $u \in X^*$ of length s .

Proof. This assertion follows immediately from the definition of the behavior of weighted automata and Theorem 4.1. \square

Theorem 4.3. *Let \mathcal{A} and \mathcal{B} be max-plus automata over the complete max-plus semiring \mathbb{R}_∞ , and let $\xi \in \mathbb{R}$. If \mathcal{A}^ξ and \mathcal{B}^ξ are either the lower or the upper ξ -shifts of \mathcal{A} and \mathcal{B} , then*

- (a) $\llbracket \mathcal{A} \rrbracket \leq \llbracket \mathcal{B} \rrbracket$ if and only if $\llbracket \mathcal{A}^\xi \rrbracket \leq \llbracket \mathcal{B}^\xi \rrbracket$;
- (b) $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{B} \rrbracket$ if and only if $\llbracket \mathcal{A}^\xi \rrbracket = \llbracket \mathcal{B}^\xi \rrbracket$.

Proof. From (21) it follows directly that $\llbracket \mathcal{A} \rrbracket \leq \llbracket \mathcal{B} \rrbracket$ implies $\llbracket \mathcal{A}^\xi \rrbracket \leq \llbracket \mathcal{B}^\xi \rrbracket$. From (21) we also obtain that $\llbracket \mathcal{A} \rrbracket(u) = (-\xi)^{[s+1]} \otimes \llbracket \mathcal{A}^\xi \rrbracket(u)$, for each word $u \in X^*$ of length s , from which it follows that $\llbracket \mathcal{A}^\xi \rrbracket \leq \llbracket \mathcal{B}^\xi \rrbracket$ implies $\llbracket \mathcal{A} \rrbracket \leq \llbracket \mathcal{B} \rrbracket$. This completes the proof of statement (a).

The statement (b) follows directly from (a). \square

Theorem 4.4. *For any $\lambda \in \mathbb{Z}$ and arbitrary max-plus automata $\mathcal{A} = (m, \sigma^A, \{M_x^A\}_{x \in X}, \tau^A)$ and $\mathcal{B} = (n, \sigma^B, \{M_x^B\}_{x \in X}, \tau^B)$ over the lower λ -truncation $\mathbb{Z}_\infty^\lambda$ of \mathbb{Z}_∞ , the sequence (19) is finite.*

Proof. First we note that the semiring $\mathbb{Z}_\infty^\lambda$ satisfies the *descending chain condition*, that is, every descending sequence in $\mathbb{Z}_\infty^\lambda$ is finite. Therefore, for any pair $(i, j) \in [1..m] \times [1..n]$, the descending sequence $\{U_s(i, j)\}_{s \in \mathbb{N}}$ is finite, so there exists a smallest member of this sequence. This means that there is a smallest number $k_{i,j} \in \mathbb{N}$ such that $U_{k_{i,j}}(i, j)$ is the smallest member of the sequence $\{U_s(i, j)\}_{s \in \mathbb{N}}$, and since the sequence is decreasing, we obtain that

$$U_{k_{i,j}}(i, j) = U_l(i, j), \quad \text{for every } l \in \mathbb{N}, l \geq k_{i,j}.$$

Now, for

$$k = \max\{k_{i,j} \mid (i, j) \in [1..m] \times [1..n]\}.$$

we have that $U_k(i, j) = U_l(i, j)$, for all $(i, j) \in [1..m] \times [1..n]$ and $l \in \mathbb{N}, l \geq k$, and consequently, $U_k = U_l$, for every $l \in \mathbb{N}, l \geq k$. This completes the proof that the sequence (19) is finite. \square

The following example demonstrates the application of lower truncations of the complete max-plus semiring.

Example 4.1. For $X = \{x, y\}$, 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 \mathbb{Z}_∞ given by

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

(a) *Computation over the complete max-plus semiring \mathbb{Z}_∞ :* Using formulas from (19), by induction on s we obtain that the members of the sequence $\{U_s\}_{s \in \mathbb{N}}$ are given by

$$U_1 = \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & -3 & 0 \end{bmatrix}, \quad U_s = \begin{bmatrix} 0 & 0 & 5 - 2s \\ -1 & -3 & 0 \end{bmatrix}, \text{ for } s \in \mathbb{N}, s \geq 2,$$

and obviously, the infimum of this sequence is

$$\widehat{U} = \begin{bmatrix} 0 & 0 & -\infty \\ -1 & -3 & 0 \end{bmatrix}.$$

As we have already said, \widehat{U} is the greatest forward presimulation between \mathcal{A} and \mathcal{B} . Since

$$\sigma^B \otimes (\widehat{U})^\top = [1 \ -\infty \ 2] \otimes \begin{bmatrix} 0 & -1 \\ 0 & -3 \\ -\infty & 0 \end{bmatrix} = [1 \ 2] > [-\infty \ 2] = \sigma^A,$$

we conclude that \widehat{U} satisfies (fs1), so it is the greatest forward simulation between \mathcal{A} and \mathcal{B} .

(b) *Computation over the lower λ -truncation $\mathbb{Z}_\infty^\lambda$ of \mathbb{Z}_∞ , for an arbitrary $\lambda \leq -3$:* By means of formulas (19) we compute the members of the sequence $\{U_s^\lambda\}_{s \in \mathbb{N}}$ and obtain

$$U_1^\lambda = \begin{bmatrix} +\infty & +\infty & +\infty \\ -\infty & -\infty & \lambda \end{bmatrix}, \quad U_2^\lambda = \begin{bmatrix} +\infty & +\infty & 1 + \lambda \\ -\infty & -\infty & \lambda \end{bmatrix}, \quad U_3 = U_4 = \widehat{U}^\lambda = \begin{bmatrix} -\infty & -\infty & -\infty \\ -\infty & -\infty & \lambda \end{bmatrix},$$

where \widehat{U}^λ denotes the infimum of this sequence. Now we have that

$$\sigma^B \otimes_\lambda (\widehat{U}^\lambda)^\top = [1 \ -\infty \ 2] \otimes_\lambda \begin{bmatrix} -\infty & -\infty \\ -\infty & -\infty \\ -\infty & \lambda \end{bmatrix} = [-\infty \ 2] = \sigma^A,$$

which means that \widehat{U}^λ satisfies (fs1) and it is the greatest forward simulation between the lower λ -shifts \mathcal{A}^λ and \mathcal{B}^λ .

(c) *Computation over the upper v -truncation \mathbb{Z}_∞^v of \mathbb{Z}_∞ , for $v = 4$:* In the case when \mathcal{A} and \mathcal{B} are regarded as weighted automata over \mathbb{Z}_∞^v , for $v = 4$, we obtain an infinite sequence

$$U_1^v = \begin{bmatrix} +\infty & +\infty & +\infty \\ 3 & 1 & 4 \end{bmatrix}, \quad U_s^v = \begin{bmatrix} 4 & 4 & 8 - 2s \\ 3 & 1 & 4 \end{bmatrix}, \text{ for } s \in \mathbb{N}, s \geq 2,$$

whose infimum is

$$\widehat{U}^v = \begin{bmatrix} 4 & 4 & -\infty \\ 3 & 1 & 4 \end{bmatrix}.$$

It is easy to verify that \widehat{U}^v satisfies (fs1), which means that it is greatest forward simulation between the upper v -shifts \mathcal{A}^v and \mathcal{B}^v .

(d) *The same automata with changed initial weights vectors:* Let us change initial weights vectors of \mathcal{A} and \mathcal{B} so that

$$\sigma^A = [1 \quad -\infty], \quad \sigma^B = [1 \quad -\infty \quad -\infty].$$

Since the initial weight vectors do not affect the computation of the elements of the sequence $\{U_s\}_{s \in \mathbb{N}}$, we obtain the same sequence as in (a) and its infimum \widehat{U} , and now we have that

$$\sigma^B \otimes (\widehat{U})^\top = [1 \quad -\infty \quad -\infty] \otimes \begin{bmatrix} 0 & -1 \\ 0 & -3 \\ -\infty & 0 \end{bmatrix} = [1 \quad 0] > [1 \quad -\infty] = \sigma^A.$$

Therefore, \widehat{U} still satisfies (fs1), and it is the greatest forward simulation between \mathcal{A} and \mathcal{B} .

When we perform computation over $\mathbb{Z}_\infty^\lambda$, for automata \mathcal{A}^λ and \mathcal{B}^λ , we also get the same sequence as in (b) and its infimum \widehat{U}^λ . However, \widehat{U}^λ no longer satisfies (fs1), since

$$\sigma^B \otimes_\lambda (\widehat{U}^\lambda)^\top = [1 \quad -\infty \quad -\infty] \otimes \begin{bmatrix} -\infty & -\infty \\ -\infty & -\infty \\ -\infty & \lambda \end{bmatrix} = [-\infty \quad -\infty] \not> [1 \quad -\infty] = \sigma^A,$$

which means that in this case there is no forward simulation between automata \mathcal{A}^λ and \mathcal{B}^λ .

Let us analyze the previous example. First, it should be pointed out that in cases where the greatest forward presimulation between two max-plus automata over \mathbb{R}_∞ cannot be computed effectively using standard algorithms and software, because the sequence of matrices that determines it is infinite, one may instead attempt to compute it over a lower truncation of \mathbb{R}_∞ . Namely, the previous example shows that even in such a case it may happen that the greatest forward presimulation between the lower shifts of these automata can be computed in a finite number of steps, that is, that the sequence of matrices defining it is finite. Moreover, Theorem 4.4 shows that this always occur for max-plus automata with integer weights. In accordance with Theorem 4.3, this can transfer the problem of determining the existence of a containment relation from max-plus automata over \mathbb{R}_∞ to their lower shifts.

However, the previous example also shows that the application of the method described above has its limitations. Namely, although a slightly modified residuum operation on $\mathbb{R}_\infty^\lambda$ can ensure that the greatest forward presimulation \widehat{U}^λ is computed over $\mathbb{R}_\infty^\lambda$ in a finite number of steps, we have seen here that such a residuum operation can cause \widehat{U}^λ to be so small that it can no longer satisfy (fs1). Therefore, it is possible to reach a situation where we have two max-plus automata between which a containment relation exists, for the original automata over \mathbb{R}_∞ it can be detected by a forward simulation that cannot be effectively computed using standard algorithms and software (except perhaps by some AI system capable of using mathematical induction), but it cannot be detected over the lower λ -truncation $\mathbb{R}_\infty^\lambda$ of \mathbb{R}_∞ , even though over $\mathbb{R}_\infty^\lambda$ the greatest forward presimulation can be computed effectively.

Finally, this example also shows that the application of upper truncations does not yield results as good as those obtained with lower truncations.

The second example, presented below, shows that even when the greatest forward presimulation between two max-plus automata over \mathbb{R}_∞ can be computed in a finite number of steps (when the corresponding sequence of matrices is finite), the use of lower truncations can still yield better results, in the sense that the computation can be carried out faster, in fewer steps.

Example 4.2. For $X = \{x, y\}$, let $\mathcal{A} = (3, \sigma^A, \{M_x^A, M_y^A\}, \tau^A)$ and $\mathcal{B} = (2, \sigma^B, \{M_x^B, M_y^B\}, \tau^B)$ be max-plus automata over \mathbb{Z}_∞ given by

$$\begin{aligned} M_x^A &= \begin{bmatrix} 10 & 3 & 4 \\ 5 & 10 & 3 \\ 4 & 6 & 7 \end{bmatrix}, & M_y^A &= \begin{bmatrix} 5 & 6 & 2 \\ 3 & 3 & 4 \\ 7 & 7 & 10 \end{bmatrix}, & \sigma^A &= [5 \ 0 \ 0], & \tau^A &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \\ M_x^B &= \begin{bmatrix} 10 & 6 \\ 6 & 7 \end{bmatrix}, & M_y^B &= \begin{bmatrix} 6 & 6 \\ 7 & 10 \end{bmatrix}, & \sigma^B &= [5 \ 0], & \tau^B &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \end{aligned}$$

(a) *Computation over the complete max-plus semiring \mathbb{Z}_∞ :* Computing the members of the sequence $\{U_s\}_{s \in \mathbb{N}}$ using formulas (19) we obtain

$$U_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad U_2 = \begin{bmatrix} 0 & -3 \\ 0 & -3 \\ -4 & 0 \end{bmatrix}, \quad U_3 = U_4 = \widehat{U} = \begin{bmatrix} 0 & -4 \\ 0 & -4 \\ -4 & 0 \end{bmatrix}$$

and since \widehat{U} satisfies (fs1), it is the greatest forward simulation between \mathcal{A} and \mathcal{B} .

(b) *Computation over the lower λ -truncation $\mathbb{Z}_\infty^\lambda$ of \mathbb{Z}_∞ , for $\lambda = 0$:* When the computation from (a) is performed over $\mathbb{Z}_\infty^\lambda$, we obtain a sequence $\{U_s^\lambda\}_{s \in \mathbb{N}}$ whose members are given by

$$U_1^\lambda = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad U_2^\lambda = U_3^\lambda = \widehat{U}^\lambda = \begin{bmatrix} 0 & -\infty \\ 0 & -\infty \\ -\infty & 0 \end{bmatrix}$$

and \widehat{U}^λ is also a forward simulation between \mathcal{A}^λ and \mathcal{B}^λ , since it satisfies (fs1).

The last theorem of this section establishes a connection between forward presimulations and simulations between max-plus automata over \mathbb{R}_∞ and forward presimulations and simulations between their lower and upper shifts.

Theorem 4.5. *Let \mathcal{A} and \mathcal{B} be max-plus automata of dimensions m and n over the complete max-plus semiring \mathbb{R}_∞ , let $\xi \in \mathbb{R}$, let \mathcal{A}^ξ and \mathcal{B}^ξ be either the lower or the upper ξ -shifts of \mathcal{A} and \mathcal{B} , and let U be an $m \times n$ matrix with entries in \mathbb{R}_∞^ξ . Then*

(a) *U is a forward presimulation between \mathcal{A}^ξ and \mathcal{B}^ξ if and only if $(-\xi) \otimes U$ is a forward presimulation between \mathcal{A} and \mathcal{B} .*

However, even if U is the greatest forward presimulation between \mathcal{A}^ξ and \mathcal{B}^ξ , $(-\xi) \otimes U$ need not be the greatest forward presimulation between \mathcal{A} and \mathcal{B} .

(b) *U is a forward simulation between \mathcal{A}^ξ and \mathcal{B}^ξ if and only if $(-\xi) \otimes U$ is a forward simulation between \mathcal{A} and \mathcal{B} .*

However, even if U is the greatest forward simulation between \mathcal{A}^ξ and \mathcal{B}^ξ , $(-\xi) \otimes U$ need not be the greatest forward simulation between \mathcal{A} and \mathcal{B} .

Proof. (a) According to Theorem 4.1 and the properties of scalar multiplication, for an arbitrary $x \in X$ we get

$$\begin{aligned} U^\top \otimes_\xi M_x^A \leq M_x^B \otimes_\xi U^\top &\Leftrightarrow (-\xi) \otimes U^\top \otimes M_x^A \leq (-\xi) \otimes M_x^B \otimes U^\top \\ &\Leftrightarrow ((-\xi) \otimes U^\top) \otimes M_x^A \leq M_x^B \otimes ((-\xi) \otimes U^\top) \\ &\Leftrightarrow ((-\xi) \otimes U)^\top \otimes M_x^A \leq M_x^B \otimes ((-\xi) \otimes U)^\top, \end{aligned}$$

and also

$$\begin{aligned} U^\top \otimes_\xi \tau^A \leq \tau^B &\Leftrightarrow (-\xi) \otimes U^\top \otimes \tau^A \leq \tau^B \Leftrightarrow ((-\xi) \otimes U^\top) \otimes \tau^A \leq \tau^B \\ &\Leftrightarrow ((-\xi) \otimes U)^\top \otimes \tau^A \leq \tau^B, \end{aligned}$$

and therefore, U is a forward presimulation between \mathcal{A}^ξ and \mathcal{B}^ξ if and only if $(-\xi) \otimes U$ is a forward presimulation between \mathcal{A} and \mathcal{B} .

Further, consider max-plus automata \mathcal{A} and \mathcal{B} over \mathbb{R}_∞ from Example 4.1. If we assume that $\xi = \lambda$ and U is the greatest forward presimulation between the lower λ -shifts \mathcal{A}^λ and \mathcal{B}^λ computed in Example 4.1 (b), that is,

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

then

$$(-\lambda) \otimes U = (-\lambda) \otimes \begin{bmatrix} -\infty & -\infty & -\infty \\ -\infty & -\infty & \lambda \end{bmatrix} = \begin{bmatrix} -\infty & -\infty & -\infty \\ -\infty & -\infty & 0 \end{bmatrix} < \begin{bmatrix} 0 & 0 & -\infty \\ -1 & -3 & 0 \end{bmatrix},$$

where the last matrix is the greatest forward presimulation between \mathcal{A} and \mathcal{B} computed in Example 4.1 (a). Thus, $(-\lambda) \otimes U$ is not the greatest forward presimulation between \mathcal{A} and \mathcal{B} .

(b) According to Theorem 4.1 and the properties of scalar multiplication, we have that

$$\begin{aligned} \sigma^A \leq \sigma^B \otimes_\xi U^\top &\Leftrightarrow \sigma^A \leq (-\xi) \otimes \sigma^B \otimes U^\top \Leftrightarrow \sigma^A \leq \sigma^B \otimes ((-\xi) \otimes U^\top) \\ &\Leftrightarrow \sigma^A \leq \sigma^B \otimes ((-\xi) \otimes U)^\top, \end{aligned}$$

and in accordance with (a), U is a forward simulation between \mathcal{A}^ξ and \mathcal{B}^ξ if and only if $(-\xi) \otimes U$ is a forward simulation between \mathcal{A} and \mathcal{B} .

The remaining part of the proof of claim (b) is contained in the proof of (a). \square

Remark 4.1. Let us once again consider the automata \mathcal{A} and \mathcal{B} from Example 4.1. As shown in part (c) of that example, for $\xi = v = 4$, the greatest forward simulation between the upper v -shifts \mathcal{A}^v and \mathcal{B}^v is

$$U = \begin{bmatrix} 4 & 4 & -\infty \\ 3 & 1 & 4 \end{bmatrix}.$$

Then

$$(-\xi) \otimes U = (-4) \otimes \begin{bmatrix} 4 & 4 & -\infty \\ 3 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\infty \\ -1 & -3 & 0 \end{bmatrix},$$

and it coincides with the greatest forward simulation between \mathcal{A} and \mathcal{B} .

Thus, if U is the greatest forward presimulation or simulation between \mathcal{A}^ξ and \mathcal{B}^ξ , then $(-\xi) \otimes U$ need not be, but may be, the greatest presimulation or simulation between \mathcal{A} and \mathcal{B} .

5. Remarks on Weak Simulations and Bisimulations for Max-Plus Automata

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

$$\sigma_u^A = \sigma^A \otimes M_u^A, \quad \tau_u^A = M_u^A \otimes \tau^A. \quad (22)$$

According to this definition, for the empty word ε we have that $\sigma_\varepsilon^A = \sigma^A$ and $\tau_\varepsilon^A = \tau^A$. The vectors σ_u^A and τ_u^A will be referred to, respectively, as the σ -vector and the τ -vector corresponding to the word $u \in X^*$.

In the context of max-plus automata, the vectors σ_u^A are known as *generalized daters* (cf. [20,30]) or *state vectors* (cf. [16,31]). The vectors σ_u^A and τ_u^A play a very important role in the determinization of max-plus automata, and recently, they have also been used to define generalizations of simulations

and bisimulations for max-plus automata, called *weak simulations* and *weak bisimulations* [33]. Namely, for max-plus automata $\mathcal{A} = (m, \sigma^A, \{M_x^A\}_{x \in X}, \tau^A)$ and $\mathcal{B} = (n, \sigma^B, \{M_x^B\}_{x \in X}, \tau^B)$, a matrix $U \in \mathbb{R}_\infty^{m \times n}$ is called a *weak forward simulation* between \mathcal{A} and \mathcal{B} if it satisfies the following conditions

$$\begin{aligned} \text{(wfs1)} \quad & \sigma^A \leq \sigma^B \otimes U^\top, \\ \text{(wfs2)} \quad & U^\top \otimes \tau_u^A \leq \tau_u^B, \text{ for all } u \in X^*, \end{aligned}$$

Weak backward simulations between \mathcal{A} and \mathcal{B} are defined as forward simulations between the reverse automata of \mathcal{A} and \mathcal{B} , and by combining the concepts of forward and backward simulations for the matrix U and its transpose U^\top , two types of bisimulations are defined. Here, the conditions (wfs1) and (wfs2) can also be treated as systems of matrix inequations with the matrix U as the unknown, and weak forward simulations can therefore be defined as the solutions of that system. Other types of weak simulations and bisimulations can also be defined as solutions of appropriate systems of matrix inequations. A matrix $U \in \mathbb{R}_\infty^{m \times n}$ that is a solution of the system composed of inequations from (wfs2) will be called a *weak forward presimulation* between \mathcal{A} and \mathcal{B} .

The procedure for testing the existence of a weak forward simulation between \mathcal{A} and \mathcal{B} , and computing the greatest one when it exists, is partly similar to the procedure described above for ordinary forward simulations, but there are also significant differences. First, the greatest solution \hat{U} of the system consisting of the matrix inequations from (wfs2) is computed, which is given by the following formula

$$\hat{U} = \bigwedge_{u \in X^*} \tau_u^A \setminus \tau_u^B. \quad (23)$$

In other words, \hat{U} is the greatest weak forward presimulation between \mathcal{A} and \mathcal{B} . If \hat{U} is a non-zero matrix and satisfies (wfs1), then \hat{U} is the greatest weak forward simulation between \mathcal{A} and \mathcal{B} . Otherwise, if \hat{U} is either the zero matrix or does not satisfy (wfs1), then there exists no weak forward simulation between \mathcal{A} and \mathcal{B} . The main problem that arises when computing the matrix \hat{U} is the possibility that the system (wfs2) consists of infinitely many inequations, which means that, before computing the matrix \hat{U} , one must compute infinitely many pairs of the form (τ_u^A, τ_u^B) , which will be called τ -pairs in \mathcal{A} . In the following example, we will see that this problem cannot be resolved by passing to the lower or upper truncations of the semiring \mathbb{R}_∞ , that is, to the corresponding lower or upper shifts of the original automata, since the sets of τ -pairs may be infinite in these cases as well.

Example 5.1. For $X = \{x\}$, 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 \mathbb{Z}_∞ given by

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

(a) *Computation over the complete max-plus semiring \mathbb{Z}_∞ :* The τ -pairs computed over \mathbb{Z}_∞ are given by

$$(\tau_{x^k}^A, \tau_{x^k}^B) = \left(\begin{bmatrix} -\infty \\ 2k \end{bmatrix}, \begin{bmatrix} 2k-1 \\ 4k-3 \\ 2k \end{bmatrix} \right), \text{ for all } k \in \mathbb{N}_0,$$

and the corresponding residuals are

$$\tau_{x^k}^A \setminus \tau_{x^k}^B = \begin{bmatrix} -\infty \\ 2k \end{bmatrix} \setminus \begin{bmatrix} 2k-1 \\ 4k-3 \\ 2k \end{bmatrix} = \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & 2k-3 & 0 \end{bmatrix}, \text{ for all } k \in \mathbb{N}_0.$$

As the sequence $2k - 3$ is increasing, its smallest member is obtained for $k = 0$ and is equal to -3 , so the greatest weak forward presimulation between \mathcal{A} and \mathcal{B} is given by

$$\widehat{U} = \bigwedge_{k \in \mathbb{N}_0} \tau_{x^k}^A \setminus \tau_{x^k}^B = \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & -3 & 0 \end{bmatrix}.$$

\widehat{U} is also the greatest weak forward simulation between \mathcal{A} and \mathcal{B} , since it satisfies (wfs1).

(b) *Computation over the lower λ -truncation $\mathbb{Z}_\infty^\lambda$ of \mathbb{Z}_∞ , for $\lambda = -3$:* When we compute the τ -pairs over $\mathbb{Z}_\infty^\lambda$ we obtain that

$$(\tau_{x^k}^A, \tau_{x^k}^B) = \left(\begin{bmatrix} -\infty \\ 5k \end{bmatrix}, \begin{bmatrix} 5k - 1 \\ 7k - 3 \\ 5k \end{bmatrix} \right), \quad \text{for all } k \in \mathbb{N}_0,$$

and the corresponding residuals are

$$\begin{aligned} \tau_\varepsilon^A \setminus \lambda \tau_\varepsilon^B &= \begin{bmatrix} +\infty & +\infty & +\infty \\ -1 & -\infty & -3 \end{bmatrix}, & \tau_x^A \setminus \lambda \tau_x^B &= \begin{bmatrix} +\infty & +\infty & +\infty \\ -\infty & -\infty & -3 \end{bmatrix}, \\ \tau_{x^k}^A \setminus \lambda \tau_{x^k}^B &= \begin{bmatrix} +\infty & +\infty & +\infty \\ -\infty & 2k - 6 & -3 \end{bmatrix}, & \text{for all } k \in \mathbb{N}, k \geq 2. \end{aligned}$$

Therefore, in this case the greatest weak forward presimulation is

$$\widehat{U}^\lambda = \bigwedge_{k \in \mathbb{N}_0} \tau_{x^k}^A \setminus \lambda \tau_{x^k}^B = \begin{bmatrix} +\infty & +\infty & +\infty \\ -\infty & -\infty & -3 \end{bmatrix},$$

and since it satisfies (wfs1), \widehat{U}^λ is the greatest weak forward simulation between \mathcal{A}^λ and \mathcal{B}^λ .

(c) *Computation over the upper v -truncation \mathbb{Z}_∞^v of \mathbb{Z}_∞ , for $v = 4$:* Computation of the τ -pairs over \mathbb{Z}_∞^v yields

$$(\tau_{x^k}^A, \tau_{x^k}^B) = \left(\begin{bmatrix} -\infty \\ -2k \end{bmatrix}, \begin{bmatrix} -2k - 1 \\ -3 \\ -2k \end{bmatrix} \right), \quad \text{for all } k \in \mathbb{N}_0,$$

and the corresponding residuals are

$$\begin{aligned} \tau_\varepsilon^A \setminus v \tau_\varepsilon^B &= \begin{bmatrix} +\infty & +\infty & +\infty \\ 4 & 1 & 4 \end{bmatrix}, & \tau_x^A \setminus v \tau_x^B &= \begin{bmatrix} +\infty & +\infty & +\infty \\ 3 & 3 & 4 \end{bmatrix}, \\ \tau_{x^k}^A \setminus v \tau_{x^k}^B &= \begin{bmatrix} +\infty & +\infty & +\infty \\ 3 & 4 & 4 \end{bmatrix}, & \text{for all } k \in \mathbb{N}, k \geq 2. \end{aligned}$$

Consequently, here we have that the greatest weak forward presimulation is

$$\widehat{U}^v = \bigwedge_{k \in \mathbb{N}_0} \tau_{x^k}^A \setminus v \tau_{x^k}^B = \begin{bmatrix} +\infty & +\infty & +\infty \\ 3 & 1 & 4 \end{bmatrix},$$

and since it satisfies (wfs1), \widehat{U}^v is the greatest weak forward simulation between \mathcal{A}^v and \mathcal{B}^v .

The algorithm proposed in [33] computes the greatest weak forward presimulation between two max-plus automata by calculating the infimum (23) step by step, simultaneously with the computation of the τ -pairs and their residuals. More precisely, in each individual step of the algorithm, a new pair $(\tau_{xu}^A, \tau_{xu}^B) = (M_x^A \otimes \tau_u^A, M_x^B \otimes \tau_u^B)$ is computed from some previously computed pair (τ_u^A, τ_u^B) , and after that, in the same step, the algorithm computes the residual $\tau_{xu}^A \setminus \tau_{xu}^B$ of that pair, as well as the infimum of this residual and the infimum of all previously computed residuals of τ -pairs. In other words,

the infimum (23) is updated at each step, and its computation terminates after all τ -pairs have been computed, provided that there are finitely many of them.

Besides the fact that passing to the computation of τ -pairs over the lower and upper truncations of the semiring \mathbb{R}_∞ does not necessarily ensure the finiteness of the set of τ -pairs, the previous example shows something else as well. Namely, in part (a) of this example, the infimum (23) is reached already in the first step, while in parts (b) and (c) it is reached in the second step, and all subsequent computations of τ -pairs and their residuals are unnecessary, since these residuals do not change the infimum that has already been reached. This naturally raises the question of whether a termination criterion can be found for such an algorithm that would recognize that the infimum has been reached and terminate the further generation of τ -pairs. Such a criterion will be the subject of our future research. Our further research will also address different methods of truncating the complete max-plus semiring, which will ensure that the sets of τ -pairs over such truncated semirings are necessarily finite.

Author Contributions: Conceptualization, J.M., M.Ć., J.I., and I.M.; methodology, M.Ć. and J.I.; software, J.M.; validation, M.Ć., J.M.; formal analysis, J.M., M.Ć., J.I., and I.M.; investigation, J.M., M.Ć., J.I., and I.M.; resources, J.M.; data curation, J.M.; writing – original draft preparation, M.Ć. and J.M.; writing – review and editing, J.M., M.Ć., J.I., and I.M.; visualization, J.M. and M.Ć.; supervision, M.Ć., J.I. and I.M.; project administration, J.M. and M.Ć.; funding acquisition, M.Ć. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science, Technological Development and Innovation, Republic of Serbia, Contract No. 451-03-137/2025-03/200124.

Data Availability Statement: The data that support the findings of this study are available on request to the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Akian M., Gaubert S., Guterman A. Tropical polyhedra are equivalent to mean payoff games. *Internat. J. Algebra Comput.* **2012**, 22(01), 1250001.
2. Baccelli, F., Cohen, G., Olsder, G., Quadrat, J. *Synchronization and Linearity*, Publisher: John Wiley and Sons Ltd, Chichester, 1992.
3. Bezem, M., Nieuwenhuis, R., Rodríguez-Carbonell, E. Exponential behaviour of the Butković–Zimmermann algorithm for solving two-sided linear systems in max-algebra. *Discrete Appl. Math.* **2008**, 156(18), 3506–3509.
4. Boukra, R., Lahaye, S., Boimond J-L. New representations for (max,+) automata with applications to the performance evaluation of discrete event systems. *IFAC PapersOnLine* **2012**, 45(29), 116–121.
5. Butković, P. *Max-linear Systems: Theory and Algorithms*, Publisher: Springer, London, 2010.
6. Butković, P., Zimmermann, K. A strongly polynomial algorithm for solving two-sided linear systems in max-algebra. *Discrete Appl. Math.* **2006**, 154(3), 437–446.
7. Cassandras, C.G., Lafortune, S. *Introduction to Discrete Event Systems*, Publisher: Springer, New York, 2008.
8. Ćirić, M., Ignjatović, J., Damljanović, N., Bašić, M. Bisimulations for fuzzy automata. *Fuzzy Sets Syst.* **2012**, 186, 100–139.
9. Ćirić, M., Ignjatović, J., Jančić, I., Damljanović, N. Computation of the greatest simulations and bisimulations between fuzzy automata. *Fuzzy Sets Syst.* **2012**, 208, 22–42.
10. Ćirić, M., Ignjatović, J., Stanimirović, P.S., Bisimulations for weighted finite automata over semirings, *Research Square*, 19 December 2022, <https://doi.org/10.21203/rs.3.rs-2386298/v1>.
11. Ćirić, M., Micić, I., Matejić, J., Stamenković, A. Simulations and bisimulations for max-plus automata. *Discrete Event Dyn. Syst.* **2024**, 34, 269–295.
12. Colcombet, T., Daviaud, L., Zuleger, F. Size-change abstraction and max-plus automata. In *Mathematical Foundations of Computer Science (MFCS 2014)*, Csuhaj-Varjú, E., Dietzfelbinger, M., Ésik, Z., Eds.; Lecture Notes in Computer Science, vol. 8634. Publishing House: Springer, Berlin-Heidelberg, 2014, pp. 208–219.
13. Damljanović, N., Ćirić, M., Ignjatović, J. Bisimulations for weighted automata over an additively idempotent semiring. *Theor. Comput. Sci.* **2014**, 534, 86–100.
14. Daviaud, L. Containment and Equivalence of Weighted Automata: Probabilistic and max-plus cases. In *Language and Automata Theory and Applications (LATA 2020)*, Leporati, A., Martín-Vide, C., Shapira, D.,

- Zandron, C., Eds.; Lecture Notes in Computer Science, vol 12038, Publishing House: Springer, Cham, 2020, pp. 17–32.
15. Daviaud, L., Guillon, P., Merlet, G. Comparison of max-plus automata and joint spectral radius of tropical matrices. In *Foundations of Computer Science (MFCS 2017)*, Larsen, K., Bodlaender, H., Raskin, J.F., Eds.; Leibniz International Proceedings in Informatics (LIPIcs), vol. 83. 2017, pp. 19:1–19:14.
 16. Daviaud, B., Lahaye, S., Lhommeau, M., Komenda, J. On the existence of simulations for max-plus automata. *IEEE Control Syst. Lett.* **2024**, *8*, 694–699.
 17. Droste, M., Kuich, W. Semirings and Formal Power Series. In *Handbook of Weighted Automata*; Droste, M., Kuich, W., Vogler, H., Eds.; Monographs in Theoretical Computer Science. An EATCS Series. Publishing House: Springer, Berlin-Heidelberg, 2009; pp. 3–28.
 18. Eklund, P., Gutiérrez García, J., Höhle, U., Kortelainen, J. *Semigroups in complete lattices*, Developments in mathematics, Publisher: Springer, Cham, 2018.
 19. Gaubert, S. Performance evaluation of $(\max,+)$ automata. *IEEE Trans. Automat. Control* **1995**, *40*(12), 2014–2025.
 20. Gaubert S., Mairesse J. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Automat. Control* **1999**, *44*(4), 683–698.
 21. Golan, J.S. *Semirings and their Applications*, Publisher: Springer, Dordrecht, 1999.
 22. Gondran, M., Minoux, M. *Graphs, Dioids and Semirings. New Models and Algorithms*, Publisher: Springer, New York, 2008.
 23. Ignjatović, J., Ćirić, M. Moore-Penrose equations in involutive residuated semigroups and involutive quantales. *Filomat* **2017**, *31*(2), 183–196.
 24. Jančić, I. Weak bisimulations for fuzzy automata. *Fuzzy Sets Syst.* **2014**, *249*, 49–72.
 25. Komenda, J., Lahaye, S., Boimond, J.L. Supervisory control of $(\max,+)$ automata: A behavioral approach. *Discrete Event Dyn. Syst.* **2009**, *19*(4), 525–549.
 26. Komenda, J., Lahaye, S., Boimond, J.-L. $(\max,+)$ -automata with partial observations. *IFAC PapersOnLine* **2018**, *51*(7), 192–197.
 27. Komenda, J., Lahaye, S., Boimond, J.L., van den Boom, T. Max-plus algebra and discrete event systems. *IFAC PapersOnLine* **2017**, *50*(1), 1784–1790.
 28. Komenda, J., Lahaye, S., Boimond, J.L., van den Boom, T. Max-plus algebra in the history of discrete event systems. *Annual Rev. Control* **2018**, *45*, 240–249.
 29. Lahaye, S., Komenda, J., Boimond, J.L. Modeling of timed Petri nets using deterministic $(\max,+)$ automata. *IFAC Proceedings Volumes* **2014**, *47*(2), 471–476.
 30. Lahaye, S., Komenda, J., Boimond, J.L. Compositions of $(\max,+)$ automata. *Discrete Event Dyn. Syst.* **2015**, *25*, 323–344.
 31. Lahaye, S., Lai, A., Komenda, J., Boimond, J.L. A contribution to the determinization of max-plus automata. *Discrete Event Dyn. Syst.* **2020**, *30*, 155–174.
 32. Lombardy, S., Mairesse, J. Max-plus automata. In *Handbook of Automata Theory, Vol 1.*, Pin, J.É., Ed.; European Mathematical Society Publishing House GmbH – EMS Press, 2021, pp. 151–188.
 33. Matejić, J., Micić, I., Ćirić, M. Weak simulations and bisimulations for max-plus automata. *Filomat* (submitted for publication).
 34. Paseka, J., Rosický, J. Quantales. In *Current Research in Operational Quantum Logic: Algebras, Categories, Languages*; Coecke, B., Moore, D., Wilce, A., Eds.; Publishing House: Springer, Dordrecht, 2000; pp. 245–262.
 35. Rosenthal, K.I. *Quantales and Their Applications*, Pitman Research Notes in Mathematics Series, Publisher: Longman Higher Education, Harlow, England, 1990.
 36. Stamenković, A., Ćirić, M., Djurdjanović, D. Weakly linear systems for matrices over the max-plus quantale. *Discrete Event Dyn. Syst.* **2022**, *32*, 1–25.
 37. Stanković, M., Ćirić, M., Ignjatović, J. Hennessy-Milner type theorems for fuzzy multimodal logics over Heyting algebras. *J. Mult.-Valued Logic Soft Comput.* **2022**, *39*(2-4), 341–379.
 38. Stanković, M., Ćirić, M., Ignjatović, J. Simulations and bisimulations for fuzzy multimodal logics over Heyting algebras. *Filomat* **2023**, *37*(3), 711–743.
 39. Triska, L., Moor, T. Behaviour equivalent max-plus automata for timed Petri nets under open-loop race-policy semantics. *Discrete Event Dyn. Syst.* **2021**, *31*, 583–607.

40. Urabe, N., Hasuo, I. Quantitative simulations by matrices. *Inf. Comput.* **2017**, *252*, 110–137.
41. Uyttendaele, J., Van Hoeck, I., Besinovic, N., Vansteenwegen, P. Timetable compression using max-plus automata applied to large railway networks, *TOP* **2023**, *31*, 414–439.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.