Logical Methods in Computer Science
Volume 21, Issue 3, 2025, pp. 3:1–3:35
https://lmcs.episciences.org/

Submitted May 03, 2024
Published Jul. 15, 2025

THE BIG-O PROBLEM FOR MAX-PLUS AUTOMATA IS DECIDABLE (PSPACE-COMPLETE)

LAURE DAVIAUD [a], DAVID PURSER [b], AND MARIE TCHENG [c]

[a] University of East Anglia, UK
e-mail address: l.daviaud@uea.ac.uk

[b] University of Liverpool, UK
e-mail address: d.purser@liverpool.ac.uk

[c] ENS Paris Saclay, France
e-mail address: marie.tcheng@ens-paris-saclay.fr

Abstract. We show that the big-O problem for max-plus automata, i.e. weighted automata over the semiring $(\mathbb{N} \cup \{-\infty\}, \max, +)$, is decidable and PSPACE-complete. The big-O (or affine domination) problem asks whether, given two max-plus automata computing functions f and g, there exists a constant c such that $f \leq cg + c$. This is a relaxation of the containment problem asking whether $f \leq g$, which is undecidable. Our decidability result uses Simon's forest factorisation theorem, and relies on detecting specific elements, that we call witnesses, in a finite semigroup closed under two special operations: stabilisation and flattening.

1. Introduction

Weighted automata are a generalisation of finite state automata, assigning values (integers, rationals, strings...) to input words, and modelling probabilities, costs or program running times. They have been introduced by Schützenberger [Sch61] and found applications in quantitative verification [CDH10] and verification of probabilistic systems [Var85], text and speech recognition [MPR02] or program complexity analysis [CDZ14].

More precisely, a weighted automaton is defined over a semiring. Commonly studied examples include the rational semiring $(\mathbb{Q}, +, \times)$ (and the particular case of probabilistic automata) and the tropical semirings $(\mathbb{N} \cup \{+\infty\}, \min, +)$ (referred to as min-plus automata) and $(\mathbb{N} \cup \{-\infty\}, \max, +)$ (max-plus automata). Non-deterministic finite automata can be viewed as weighted automata over the Boolean semiring. In all these cases, input words are mapped to rational values (and possibly $+\infty$ or $-\infty$).

Key words and phrases: Max-plus automata, Big-O, Decidability.

The first author has been supported for this work by the EPSRC grant EP/T018313/1. The second author was partially supported by the ERC grant INFSYS, agreement no. 950398. A preliminary version appeared in [DP23].

LOGICAL METHODS IN COMPUTER SCIENCE
DOI:10.46298/LMCS-21(3:3)2025

Comparing the functions computed by two given weighted automata is then a fundamental question. It is natural to consider the equivalence problem (are they equal?), and the containment problem (is one pointwise smaller than the other?). These two problems have been extensively studied and solutions are highly dependent on the semiring under consideration. Results for the equivalence problem are contrasting, but the containment problem is usually difficult:

- For the Boolean semiring, the equivalence and containment problems correspond to the language equivalence and language inclusion respectively. Both problems are PSPACE-complete [SM73].
- For the rational semiring the equivalence problem is decidable [Sch61], even in polynomial time [KMO$^+$13], but containment is undecidable [Paz14], even in restricted subclasses [DJL$^+$21].
- For the tropical semirings both problems are undecidable [Kro94]. See [ABK22, ABK11] for a comprehensive overview of the decidability boundary for min-plus automata.

In this paper we consider a relaxation of the containment problem, called the big-O problem which asks whether an automaton $\mathcal{A}$ is big-O of an automaton $\mathcal{B}$, that is, if there exists a constant c such that:

$$[\![\mathcal{A}]\!](w) \leq c[\![\mathcal{B}]\!](w) + c \text{ for all words } w$$

where $[\![\mathcal{A}]\!]$ (resp. $[\![\mathcal{B}]\!]$) denotes the function computed by $\mathcal{A}$ (resp. $\mathcal{B}$). Intuitively the problem asks whether, asymptotically, $[\![\mathcal{B}]\!]$ grows at least as fast as $[\![\mathcal{A}]\!]$ on every sequence of words.

Chistikov, Kiefer, Murawski and Purser [CKMP20, CKMP22] study the big-O problem over the non-negative rational semiring, where it is shown to be undecidable in general, but decidable for certain restrictions on the ambiguity or the accepted language. Similarly, in [CLM$^+$22], two restrictions of the big-O problem, also known to be undecidable in general, are studied on specific subclasses: the boundedness problem (a.k.a. limitedness), and the zero isolation problem.[1]

For the tropical semirings, the big-O problem has also been proved to be decidable in the $(\mathbb{N} \cup \{+\infty\}, \min, +)$ setting via the study of another problem: the domination problem [Col07, Col09]. This later asks whether there is a function $\alpha : \mathbb{N} \to \mathbb{N}$ such that $[\![\mathcal{A}]\!] \leq \alpha \circ [\![\mathcal{B}]\!]$. Affine domination requires that α be affine, and is equivalent to the big-O problem that we consider. Colcombet and Daviaud [CD13] show that domination and affine domination are equivalent and decidable for min-plus automata. More specifically, it turns out that if some function α exists then an affine α suffices. This result superseded the decidability of the boundedness problem for min-plus automata [Has91, Leu88, Sim94].

In this paper, we turn our attention to $(\mathbb{N} \cup \{-\infty\}, \max, +)$ for which the (un)decidability of the big-O problem was open. First, note that there is no obvious way to use the results obtained for min-plus automata. The natural transformation - given f computed by a min-plus automaton, $-f$ is computed by a max-plus automaton - does not preserve positivity, and the standard way to go back to $\mathbb{N}$ implies adding a big enough function to $-f$ which does not preserve the growth rate. In fact, we prove that the equivalence between domination and affine domination does not hold any more for max-plus automata (see Running Example 2). Second, the boundedness problem in this case is trivially decidable and does not provide any

[1]The boundedness problem is the special case of the big-O problem when $[\![\mathcal{B}]\!] = 1$. The zero isolation problem is the special case when $[\![\mathcal{A}]\!] = 1$. Chistikov et al. [CKMP22] deal with the problem whether $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!]$ - so without $+c$, but the two problems are equivalent (see Remark 3.6).

help. The problem for max-plus automata requires individual attention and the introduction of new tools. We show that it is PSPACE-complete and our proof provides new insights in the description of the behaviour of these automata. In [CDZ14], some description of the asymptotic behaviour of the functions computed by max-plus automata is given, but this is somehow orthogonal to the big-O problem. While providing a precise description, it is not sufficient to solve the big-O problem and new techniques are required.

Building on some standard techniques, in particular Simon's factorisation forest theorem and the stabilisation operation [Sim90, Col07, Kuf08, KL09, FGKO15, CD13, CLM+22], we construct a finite semigroup closed under the stabilisation operation and a new *flattening* operation. The stabilisation operation identifies unbounded behaviour, while the flattening operation identifies maximal growth rates. The problem reduces to detecting the presence of *witnesses* in this semigroup. A naïve search through the semigroup gives decidability, but may require exponential space. The PSPACE complexity comes from searching witnesses of a particular shape only requiring polynomial space. The hardness comes from the PSPACE-hardness of the universality problem for Boolean automata.

Organisation of the paper. The paper expands on [DP23], giving full proofs and explanations. In Section 2, we give the definition of max-plus automata and introduce a running example that we will use all along the paper. In Section 3, we state the big-O problem and a simplified version of it, prove their PSPACE-hardness and reduce our result to prove that the simplified big-O problem is in PSPACE (Theorem 3.2). In Section 4, we give a high-level description of this proof and define the semigroups that will be used and the stabilisation and flattening operations. In Section 5, we define witnesses and give a decision procedure that we show to be PSPACE. Sections 6, 7, 8 and 9 are then dedicated to prove that the decision procedure is sound and complete. The organisation and content of these sections are explained at the end of Section 5, when the suitable notions have been introduced. Finally, in Section 10, we provide a sequence of max-plus automata that require increasingly intricate witnesses (for a notion defined in the paper); this section is new compared with [DP23].

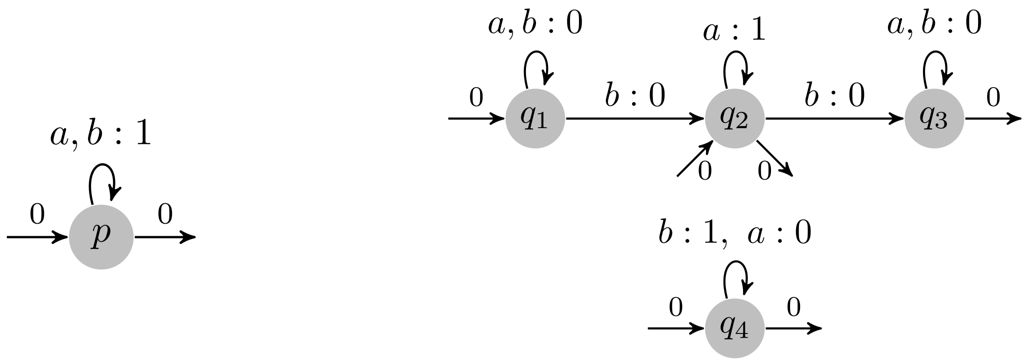
2. Max-plus automata

Let $\mathbb{N}_{\max}$ denote the set $\mathbb{N} \cup \{-\infty\}$ and note that $(\mathbb{N}_{\max}, \max, +)$ is a semiring. For some positive integers i, j, let $\mathcal{M}^{i\times j}$ denote the set of matrices of dimension $i \times j$ with coefficients in $\mathbb{N}_{\max}$. We define the product of matrices as usual on a semiring, provided the numbers of columns of A matches the number of rows of B: $(A \otimes B)_{q,q'} = \max_{q''} \left(A_{q,q''} + B_{q'',q'}\right)$. We will use the symbol $\otimes$ to denote the product of matrices on several semirings, but the context will always clearly identify which one.

Definition 2.1. A *max-plus automaton* is a tuple $\langle Q, \Sigma, M, I, F\rangle$ where Q is a finite set of states (and $|Q|$ denotes the number of states), Σ is a finite alphabet, $M : \Sigma \to \mathcal{M}^{|Q|\times|Q|}$ maps each letter to a matrix, I is a row vector in $\mathcal{M}^{1\times|Q|}$ and F a column vector in $\mathcal{M}^{|Q|\times 1}$. Moreover, the automaton is said to be deterministic if I has at most one entry different from $-\infty$ and for all a in Σ, every row of $M(a)$ has at most one entry different from $-\infty$.

Given a max-plus automaton $\langle Q, \Sigma, M, I, F\rangle$, we extend M by morphism to Σ^*.

Definition 2.2. The *weighting function* computed by $\mathcal{A} = \langle Q, \Sigma, M, I, F\rangle$, a max-plus automaton, is defined as the function $[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{N}_{\max}$ mapping a word $w = w_1 w_2 \dots w_k$,



$\mathcal{A}$: Computes word length $|w|$.

$\mathcal{B}$: Computes the maximum between the longest block of a's and the number of b's.

Figure 1. Running examples

where $w_i \in \Sigma$ for all $i = 1, \dots, k$, to:

$$[\![\mathcal{A}]\!](w) = I \otimes M(w_1) \otimes M(w_2) \otimes \cdots \otimes M(w_k) \otimes F.$$

These definitions can be expressed in terms of graphs as usual, and we will rather use the usual automaton vocabulary (transitions, runs, accepting runs, initial and final states, etc.) when appropriate in some proofs. We will often write $p \xrightarrow{w:x} q$ for a run from state p to state q labelled by the word w with weight $x \neq -\infty$, the weight of a run being the sum of the weights of the transitions in the run. In matrix terms, this means that, for $w = w_1 w_2 \dots w_k$, where $w_i \in \Sigma$ for all $i = 1, \dots, k$, there are $p = j_0, \dots, j_k = q$ such that $M(w_i)_{j_{i-1}, j_i} = x_i$ and $x = x_1 + \dots + x_k$. The run is accepting if p is initial and q is final, i.e. $I_p \neq -\infty$ and $F_q \neq -\infty$ and $[\![\mathcal{A}]\!](w)$ is equal to the maximum of the weights of the accepting runs labelled by w.

We assume that all the states in the automata under consideration in this paper are accessible and co-accessible.

The size of an automaton is the number of bits required to encode M, I and F which is bounded by $(|\Sigma| \cdot |Q|^2 + 2|Q|) \cdot \lceil \log(\Lambda) \rceil$, where Λ is the maximal weight appearing in an entry of M, I or F.

Throughout the paper we will illustrate the results and proofs using a running example that we detail now.

Running Example, Part 1. Let us consider two example automata $\mathcal{A}$ and $\mathcal{B}$ over $\Sigma = \{a, b\}$, depicted in Fig. 1. $\mathcal{A}$ computes the length of the input word and $\mathcal{B}$ computes the maximum between the length of the longest block of consecutive a's, and the number of b's.

In the following matrix descriptions, to avoid cluttering notation, we use $-$ instead of $-\infty$ to denote that there is no path.

Formally, the automaton $\mathcal{A}$ is defined by a single state $Q_{\mathcal{A}} = \{p\}$, alphabet $\Sigma = \{a, b\}$ and $M_{\mathcal{A}}(a) = (1), M_{\mathcal{A}}(b) = (1)$ with $I_{\mathcal{A}} = F_{\mathcal{A}} = (0)$.

Formally $\mathcal{B}$ is defined by states $Q_{\mathcal{B}} = \{q_1, q_2, q_3, q_4\}$, alphabet $\Sigma = \{a, b\}$ and

$$M_{\mathcal{B}}(a) = \begin{pmatrix} 0 & - & - & - \\ - & 1 & - & - \\ - & - & 0 & - \\ - & - & - & 0 \end{pmatrix} \text{ and } M_{\mathcal{B}}(b) = \begin{pmatrix} 0 & 0 & - & - \\ - & - & 0 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}$$

with $(I_\mathcal{B})_{q_1} = (I_\mathcal{B})_{q_2} = (I_\mathcal{B})_{q_4} = (F_\mathcal{B})_{q_2} = (F_\mathcal{B})_{q_3} = (F_\mathcal{B})_{q_4} = 0$, and the unspecified entries of I, F are $-\infty$. ⌟

3. Decidability of the big-O problem

The big-O problem for max-plus automata asks whether, given two max-plus automata $\mathcal{A}, \mathcal{B}$ on the same alphabet Σ, there is a positive integer c such that for all words w in Σ^*, $[\![\mathcal{A}]\!](w) \leq c[\![\mathcal{B}]\!](w) + c$. In this case, we say that $\mathcal{A}$ is big-O of $\mathcal{B}$.

Running Example, Part 2. Observe that $\mathcal{A}$ is not big-O of $\mathcal{B}$, since $[\![\mathcal{A}]\!]((a^{n-1}b)^n) = n^2$, while $[\![\mathcal{B}]\!]((a^{n-1}b)^n) = n$ for all positive integers n.

However, $[\![\mathcal{B}]\!](w) \leq [\![\mathcal{A}]\!](w)$ for all words w, so $\mathcal{B}$ is big-O of $\mathcal{A}$.

Note also that $[\![\mathcal{A}]\!] \leq ([\![\mathcal{B}]\!] + 1)[\![\mathcal{B}]\!]$, hence $[\![\mathcal{A}]\!]$ is dominated by $[\![\mathcal{B}]\!]$ as explained in the introduction, but not big-O, showing the discrepancy between the min-plus and the max-plus cases. ⌟

The main contribution of this paper is the following result.

Theorem 3.1. *The big-O problem for max-plus automata is decidable and is PSPACE-complete.*

The rest of this paper is devoted to prove this theorem. The first step is to make a number of simplifications on the automata taken as input.

Theorem 3.2. *The following problem, called the* simplified big-O problem*, is PSPACE-complete:*

- *Input: Max-plus automata $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A}$ is deterministic and $[\![\mathcal{B}]\!] : \Sigma^* \to \mathbb{N}$.*
- *Output: Yes if and only if $\mathcal{A}$ is big-O of $\mathcal{B}$.*

Compared to the big-O problem, the simplified big-O problem requires that $\mathcal{A}$ be deterministic and no word w has $[\![\mathcal{B}]\!](w) = -\infty$.

Running Example, Part 3. Recall $\mathcal{A}, \mathcal{B}$ from the running example in Running Example, Part 1. Observe that $\mathcal{A}$ is deterministic and $[\![\mathcal{B}]\!](w) \geq 0$ for all w, so $\mathcal{A}$ and $\mathcal{B}$ form an instance of the simplified big-O problem. ⌟

Proposition 3.3. *If the simplified big-O problem is decidable in PSPACE then the big-O problem is decidable in PSPACE.*

Proof. We reduce an instance of the big-O problem to the simplified big-O problem. The instance of simplified big-O problem will be of polynomial size with respect to the initial input, but a PSPACE pre-processing step is also used.

- Let $\mathcal{A}, \mathcal{B}$ be max-plus automata. Let us first construct $\mathcal{A}', \mathcal{B}'$ with $[\![\mathcal{B}']\!] : \Sigma^* \to \mathbb{N}$ such that $\mathcal{A}$ is big-O of $\mathcal{B}$ if and only if $\mathcal{A}'$ is big-O of $\mathcal{B}'$. Let $L_\mathcal{A}$ (resp. $L_\mathcal{B}$) be the (rational) language of words w such that $[\![\mathcal{A}]\!](w) \neq -\infty$ (resp. $[\![\mathcal{B}]\!](w) \neq -\infty$). Checking whether $L_\mathcal{A} \subseteq L_\mathcal{B}$ can be done in PSPACE (consider the Boolean automata obtained from $\mathcal{A}$ and $\mathcal{B}$ by ignoring the weights - they accept $L_\mathcal{A}$ and $L_\mathcal{B}$, and inclusion of rational languages is PSPACE [MS72]). If $L_\mathcal{A}$ is not included in $L_\mathcal{B}$ then $\mathcal{A}$ cannot be big-O of $\mathcal{B}$. Take $\mathcal{A}'$ that computes the length of the words and $\mathcal{B}'$ that computes the function 0. If $L_\mathcal{A}$ is included in $L_\mathcal{B}$, take $\mathcal{A}' = \mathcal{A}$ and $\mathcal{B}'$ being $\mathcal{B}$ augmented with a new state which is both initial and finial and has a self loop on all letters with weight 0. In both cases, $\mathcal{A}$ is big-O of $\mathcal{B}$ if and only if $\mathcal{A}'$ is big-O of $\mathcal{B}'$ and $[\![\mathcal{B}']\!] : \Sigma^* \to \mathbb{N}$.

- We now reduce this to the case where automaton $\mathcal{A}$ is deterministic. Consider two automata $\mathcal{A} = \langle Q_\mathcal{A}, \Sigma, M_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A}\rangle$ and $\mathcal{B} = \langle Q_\mathcal{B}, \Sigma, M_\mathcal{B}, I_\mathcal{B}, F_\mathcal{B}\rangle$ with $[\![\mathcal{B}]\!] : \Sigma^* \to \mathbb{N}$. We construct $\mathcal{A}', \mathcal{B}'$ with $\mathcal{A}'$ deterministic and $[\![\mathcal{B}']\!] : \Sigma^* \to \mathbb{N}$ such that $\mathcal{A}$ is big-O of $\mathcal{B}$ if and only if $\mathcal{A}'$ is big-O of $\mathcal{B}'$. Automata $\mathcal{A}'$ and $\mathcal{B}'$ are over the alphabet $\Sigma' = \{a_q \mid a \in \Sigma,\ q \in Q_\mathcal{A}\}$ and:
  - $\mathcal{A}'$ is constructed from $\mathcal{A}$, with states $Q_\mathcal{A} \cup \{r\}$ with r a new state (the new unique initial state), $(I_{\mathcal{A}'})_r = 0$ and all the other entries of $I_{\mathcal{A}'}$ being $-\infty$, final states $(F_{\mathcal{A}'})_p = (F_\mathcal{A})_p$ for $p \in Q_\mathcal{A}$ and $(F_{\mathcal{A}'})_r = \max\{(F_\mathcal{A})_p \mid p \text{ initial in } \mathcal{A}\ \}$, a transition $p \xrightarrow{a_q:x} q$ for each transition $p \xrightarrow{a:x} q$ in $\mathcal{A}$, and a transition $r \xrightarrow{a_q:x+(I_\mathcal{A})_p} q$ for each transition $p \xrightarrow{a:x} q$ in $\mathcal{A}$. Note that $\mathcal{A}'$ is deterministic.
  - $\mathcal{B}'$ is constructed from $\mathcal{B}$, with states $Q_\mathcal{B}$, initial and final states $I_\mathcal{B}$ and $F_\mathcal{B}$ respectively, and a transition $p \xrightarrow{a_r:x} q$ for each transition $p \xrightarrow{a:x} q$ in $\mathcal{B}$ and each $r \in Q_\mathcal{A}$. Note that $[\![\mathcal{B}]\!] : \Sigma^* \to \mathbb{N}$.

  Suppose that $\mathcal{A}$ is big-O of $\mathcal{B}$ and let us prove that $\mathcal{A}'$ is big-O of $\mathcal{B}'$. Let w be a word over Σ' and $\bar{w}$ over Σ defined as w where the subscripts of the letters are removed. Then $[\![\mathcal{A}']\!](w) \le [\![\mathcal{A}]\!](\bar{w})$ and $[\![\mathcal{B}]\!](\bar{w}) = [\![\mathcal{B}']\!](w)$ by construction and hence $\mathcal{A}'$ is big-O of $\mathcal{B}'$. Conversely, suppose that $\mathcal{A}'$ is big-O of $\mathcal{B}'$. Let $w \in \Sigma^*$. Consider an accepting run in $\mathcal{A}$ labelled by w that has maximal weight and $q_0, q_1, q_2, \ldots, q_k$ the corresponding sequence of states. Let $\bar{w} = a^1_{q_1} \cdots a^k_{q_k}$ word over Σ' where $w = a^1 \cdots a^k$ and $a^i \in \Sigma$ for all i. Then $[\![\mathcal{A}]\!](w) = [\![\mathcal{A}']\!](\bar{w})$ and $[\![\mathcal{B}']\!](\bar{w}) = [\![\mathcal{B}]\!](w)$ by construction. Hence $\mathcal{A}$ is big-O of $\mathcal{B}$. □

Proposition 3.4. *The simplified big-O problem is PSPACE-hard.*

Proof. We reduce from the Cofiniteness problem, which asks whether the language of a non-deterministic finite automaton is cofinite, that is, accepts all but a finite set of words.

Lemma 3.5. Cofiniteness *is PSPACE-hard*

Proof. We reduce from the universality of non-deterministic finite automata. Let $\mathcal{A}$ be an instance of universality over alphabet Σ; we construct $\mathcal{A}'$ over $\Sigma \cup \{\#\}$ where $\#$ is a new letter not in Σ. $\mathcal{A}'$ is such that for all words w over Σ^* and for all words u over $\Sigma \cup \{\#\}$, $w\#u$ is accepted by $\mathcal{A}'$ if and only if w is accepted in $\mathcal{A}$. This is achieved by augmenting $\mathcal{A}$ with an accepting sink state reachable on $\#$ from every accepting state of $\mathcal{A}$. We observe $\mathcal{A}$ is universal if and only if $\mathcal{A}'$ is cofinite. In particular, if $\mathcal{A}$ is universal, so too is $\mathcal{A}'$, and in particular, cofinite. If $\mathcal{A}$ does not accept w then $\mathcal{A}$ does not accept the language $w\#\Sigma^*$, and thus is not cofinite. □

Let $\mathcal{B}$ be an input to Cofiniteness; we construct an instance of the simplified big-O problem. Let $\mathcal{A}'$ and $\mathcal{B}'$ such that $[\![\mathcal{A}']\!](w) = |w|$ for all $w \in \Sigma^*$ and $[\![\mathcal{B}']\!](w) = |w|$ for all w accepted by $L_\mathcal{B}$ and 0 otherwise. The automaton $\mathcal{B}'$ is constructed from $\mathcal{B}$ by associating every edge in $\mathcal{B}$ with the weight 1 and augmented with a new state which is both initial and finial and has a self loop on zero for all letters.

We observe $\mathcal{B}$ is cofinite if and only if $\mathcal{A}$ is big-O of $\mathcal{B}$.

- If $\mathcal{B}$ is cofinite then there exists a longest word w_L that is not recognised by $\mathcal{B}$ and $[\![\mathcal{A}']\!](w) \le [\![\mathcal{B}']\!](w) + |w_L|$ for all $w \in \Sigma^*$ and $\mathcal{A}$ is big-O of $\mathcal{B}$.
- If $\mathcal{B}$ is not cofinite then there is an infinite sequence of words not accepted by $\mathcal{B}$ and in particular one of increasing length words, $(w_i)_{i\in\mathbb{N}}$. Hence we have $[\![\mathcal{A}']\!](w_i) \to \infty$ while $[\![\mathcal{B}']\!](w_i) = 0$ as $i \to \infty$ and $\mathcal{A}'$ is not big-O of $\mathcal{B}'$. □

Proposition 3.3 states that it is enough to prove that the simplified big-O problem is in PSPACE in order to conclude the proof of Theorem 3.1. Proposition 3.4 gives the hardness parts of Theorems 3.1 and 3.2 (since the simplified problem is a particular instance of the general one).

The rest of the paper, Section 4 and beyond, will then focus on proving that the simplified big-O problem is PSPACE.

Remark 3.6. Chistikov, Kiefer, Murawski and Purser [CKMP22] use a slightly different notion of big-O for rational automata, requiring the existence of c such that $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!]$ (that is, without $+c$). We note that for max-plus automata these two problems are equivalent, they reduce to each other in PSPACE, so without further blow up in complexity. Indeed, $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!] + c$ if and only if $[\![\mathcal{A}]\!] \leq c([\![\mathcal{B}]\!] + 1)$, for which the translation can be constructed in polynomial time. Conversely, there exists c such that $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!]$ if and only if there is c such that $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!] + c$ and $\{w \mid [\![\mathcal{A}]\!](w) \geq 1 \text{ and } [\![\mathcal{B}]\!](w) = 0\}$ is empty. The latter check can be done with an emptiness test for regular languages in PSPACE.

Remark 3.7. One could be interested in computing the optimal or minimal constant c such that $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!] + c$, or with the previous remark such that $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!]$. Such a c is not computable. If one could compute it and check whether it is at most 1, it could be decided whether $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$, which is undecidable for max-plus automata [Kro94].

4. Projective semigroups

From now on, we fix a deterministic max-plus automaton $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, M_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}} \rangle$ and a max-plus automaton $\mathcal{B} = \langle Q_{\mathcal{B}}, \Sigma, M_{\mathcal{B}}, I_{\mathcal{B}}, F_{\mathcal{B}} \rangle$ over the same alphabet Σ.

4.1. **Proof schema.** We describe here the general idea of the proof: this is a very high level explanation, omitting a lot of technical (but necessary) details, so should be read as such.

We want to show that either $\mathcal{A}$ is big-O of $\mathcal{B}$, or exhibit an infinite sequence of words witnessing that this is not the case. The general idea of the proof follows the (now rather standard) scheme of finding a finite semigroup, computable from $\mathcal{A}$ and $\mathcal{B}$, and identifying special elements in it - which we will call witnesses - such that witnesses exist in the semigroup associated with $\mathcal{A}$ and $\mathcal{B}$ if and only if $\mathcal{A}$ is *not* big-O of $\mathcal{B}$. To obtain the complexity bound, we will show that testing the existence of such a witness can be done in PSPACE. Similar proof schema can be found in [Sim94, CD13, CDZ14] - in particular this has been used for deciding the boundedness of distance automata, and we will explain the new insights our proof provides in comparison.

To construct the appropriate semigroup, the first step is to project the weights into $\{-\infty, 0, 1\}$. Indeed, the exact weights are not important, the only thing that matters is the difference in growth rates as illustrated in Running Example, Part 2. This leads to introduce the so-called semigroup of paths (denoted $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$) in Definition 4.3, which is finite and contains elements (p, x, q, M) where p, q are states in $\mathcal{A}$, x is in $\{0, 1\}$ and there is a word w such that there is a run in $\mathcal{A}$ from p to q labelled by w with weight 0 if $x = 0$, and positive weight if $x = 1$. Moreover, M is the matrix $M_{\mathcal{B}}(w)$ where all the (strictly) positive entries are replaced by 1. This semigroup is easily computed starting from the letters and closing under an appropriate product.

This semigroup witnesses the existence of runs with 0 or positive weights, but is not enough to compare the growth rates in $\mathcal{A}$ and $\mathcal{B}$. The next step is to add a value ∞: an entry will be ∞ if there is a sequence of words with paths with unbounded weights for this entry.

We introduce the semigroup of asymptotic behaviour (denoted $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$) in Definition 4.6, which contains elements (p, x, q, M) where p, q are states in $\mathcal{A}$, x is in $\{0, 1, \infty\}$ and M is a matrix with entries in $\{-\infty, 0, 1, \infty\}$. We introduce a first operation - the stabilisation operation (as defined in Definition 4.4) - which essentially iterates a word many times and puts value ∞ in the entries that are unbounded as the word is repeated. If starting from the letters and closing under an appropriate product and the stabilisation operation, we would get the following: for an element (p, x, q, M) in the semigroup of asymptotic behaviour, there is a sequence of words $(w_i)_i$ labelling paths in $\mathcal{A}$ from p to q with weights 0 if $x = 0$, positive but bounded weights if $x = 1$, unbounded weights if $x = \infty$. Moreover, an entry of M is 0 if it is 0 in all $M_{\mathcal{B}}(w_i)$, is 1 if it is positive and bounded for the $M_{\mathcal{B}}(w_i)$ and ∞ if it tends to $+\infty$ when $i \to +\infty$ in the matrix $M_{\mathcal{B}}(w_i)$.

Up to now the semigroup of asymptotic behaviour witnesses unboundedness of sequences of words, but not yet the difference in growth rates in $\mathcal{A}$ and $\mathcal{B}$. This is where our proof gives new insight. We introduce a second operation, which we call flattening, and behaves as follows: consider an element (p, x, p, M) in the semigroup of asymptotic behaviour such that $x = \infty$, so witnessing some sequence of words with unbounded weights in $\mathcal{A}$ from p to p. We would like to iterate the words in this sequence, no longer looking at unboundedness, but rather at which entries in M will grow linearly as the elements of the sequence are repeated. We define the flattening operation to do exactly that. After flattening, we obtain an element (p, x, p, M') witnessing a sequence of words $(w_i)_i$ such that it is possible for an entry in M' to have value 1, corresponding to paths with unbounded weights, but only if the asymptotic growth rate of these weights is little-o of the respective weights in $\mathcal{A}$ from p to p. In other words, unbounded runs of maximal growth rate continue to be represented by ∞, but runs which are not of the maximal growth rate are 'flattened' back to 1, even if they are unbounded.

The semigroup of asymptotic behaviour can be easily computed starting from the letters, closing under an appropriate product and stabilisation and flattening operations. Witnesses are those elements (p, x, q, M) where $x = \infty$, with p initial and q final in $\mathcal{A}$, and all the entries of M between an initial and a final state in $\mathcal{B}$ have value at most 1. We will use Simon's factorisation theorem [Sim90] (see Theorem 6.1) in two ways. In the case where there is no witness, this will allow us to bound a constant c such $[\![\mathcal{A}]\!] \leq c[\![\mathcal{B}]\!] + c$. In the case where there is a witness, this will allow us to limit our search to a specific type - called tractable witness of non-domination (Definition 5.2) - ensuring the PSPACE complexity of our algorithm.

Remark 4.1. We could have introduced a single operation, somehow defined as a stabilisation in some cases, and as flattening in others. We chose not to as the definition felt ad-hoc and we believe the two operations of stabilisation and flattening are more intuitive. This might have slightly simplified parts of the proof, however an added benefit we gained with two operations is to characterise exactly the shape of these special witnesses we define: the tractable witnesses of non-domination.

4.2. **Semigroup of paths.**
Projection in $\{-\infty, 0, 1, \infty\}$. Let Ω be the semiring $(\{-\infty, 0, 1, \infty\}, \max, +)$ where the operations are defined as follows: the max operation is given by the order $-\infty < 0 < 1 < \infty$ and the sum operation is commutative and given by $-\infty + x = -\infty$ for any element x (including $x = \infty$), $0 + x = x$ for any $x \in \{0, 1, \infty\}$, $1 + 1 = 1$ and $1 + \infty = \infty + \infty = \infty$.

Let $\mathcal{M}_\Omega^{i\times j}$ be the set of matrices of size $i \times j$ over the semiring Ω. We use again $\otimes$ to denote the product of matrices induced by the operations in Ω.

Given a finite set Q and a positive integer i, we denote by $(\mathfrak{M}_{Q,i}, \otimes)$ the semigroup where:

- $\mathfrak{M}_{Q,i}$ is the union of the sets $Q \times \Omega \times Q \times \mathcal{M}_\Omega^{i\times i}$ and $\{\perp\}$.
- $(p, x, q, M) \otimes (p', x', q', M') = (p, x + x', q', M \otimes M')$ if $q = p'$ and $\perp$ otherwise.

We will often denote the product of two elements $e, e' \in \mathfrak{M}_{Q,i}$ as ee' instead of $e \otimes e'$.

Projection in $\{-\infty, 0, 1\}$. We denote $\overline{-\infty} = -\infty$, $\overline{0} = 0$ and for any positive integer x, $\overline{x} = \overline{\infty} = 1$. For a matrix M in $\mathcal{M}^{i\times j}$, we denote by $\overline{M}$ the matrix M where the coefficients are replaced by their barred version.

Let $\overline{\Omega}$ be the semiring $(\{-\infty, 0, 1\}, \max, +)$ where the operations are defined as follows: the max operation is given by the order $-\infty < 0 < 1$ and the sum operation is commutative and given by $-\infty + x = -\infty$ for any element x, $0 + x = x$ for any $x \in \{0, 1\}$ and $1 + 1 = 1$.

Let $\mathcal{M}_{\overline{\Omega}}^{i\times j}$ be the set of matrices of size $i \times j$ over the semiring $\overline{\Omega}$. We use again $\otimes$ to denote the product of matrices induced by the operations in $\overline{\Omega}$.

Note that $x \mapsto \overline{x}$ is a morphism over $\mathbb{N}_{\max}$, as well as over $\mathcal{M}_\Omega^{i\times j}$. We will use the following basic properties without referencing them in the rest of the paper.

Lemma 4.2.
Let $a, b \in \mathbb{N}_{\max}$, and let $\overline{a}, \overline{b}$ be the projection of a and b into $\overline{\Omega}$, we have:
(1) *If $a \leq b$ then $\overline{a} \leq \overline{b}$.*
(2) *$\overline{a + b} = \overline{a} + \overline{b}$, where $\overline{a} + \overline{b}$ is taken in the $\overline{\Omega}$ semiring.*

Let $M, N \in (\mathbb{N}_{\max})^{d\times d}$ and let $\overline{M}, \overline{N}$ be the pointwise projection of M and N into $\overline{\Omega}^{d\times d}$, we have:
(3) *$\overline{M \otimes N} = \overline{M} \otimes \overline{N}$ where $\overline{M} \otimes \overline{N}$ is taken in the $\overline{\Omega}$ semiring.*

Proof.
(1)
- Suppose $a = -\infty$ then $-\infty = a \leq b$ and $-\infty = \overline{a} \leq \overline{b}$.
- Suppose $a = 0$, we have $a \leq b \iff b \in \mathbb{N}$. Then $\overline{a} \leq \overline{b}$.
- Suppose $a \in \mathbb{N}$, we have $a \leq b \iff b \in \mathbb{N}_{\geq a}$, thus $\overline{b} = 1$. Then $1 = \overline{a} \leq \overline{b}$.

(2)
- Suppose $a = -\infty$ (or, $b = -\infty$) then $a + b = \overline{a + b} = \overline{a} + \overline{b} = -\infty$.
- Suppose $a = b = 0$, then $a + b = \overline{a + b} = \overline{a} + \overline{b} = 0$.
- Suppose $a \in \mathbb{N}_{\geq 1}$, $b \in \mathbb{N}$. Then $a + b \in \mathbb{N}_{\geq 1}, \overline{a + b} = 1, \overline{a} = 1, \overline{a} + \overline{b} = 1$.

(3) We have
$$\begin{aligned}
\overline{(M \otimes N)_{i,k}} &= \overline{M_{i,j} + N_{j,k}} \text{ for some } j \\
&= \overline{M_{i,j}} + \overline{N_{j,k}} \\
&\leq \max_j \overline{M_{i,j}} + \overline{N_{j,k}} \\
&= (\overline{M} \otimes \overline{N})_{i,k}
\end{aligned}$$

and

$$\begin{aligned}(\overline{M} \otimes \overline{N})_{i,k} &= \overline{M_{i,j}} + \overline{N_{j,k}} \text{ for some } j \\ &= \overline{M_{i,j} + N_{j,k}} \\ &\leq \overline{\max_j M_{i,j} + M_{j,k}} \\ &= \overline{(M \otimes N)_{i,k}}.\end{aligned}$$ □

Analogously to $(\mathfrak{M}_{Q,i}, \otimes)$, given a finite set Q and a positive integer i, we denote by $(\overline{\mathfrak{M}_{Q,i}}, \otimes)$ the semigroup where:

- $\overline{\mathfrak{M}_{Q,i}}$ is the union of the sets $Q \times \overline{\Omega} \times Q \times \mathcal{M}_{\overline{\Omega}}^{i \times i}$ and $\{\perp\}$.
- $(p, x, q, M) \otimes (p', x', q', M') = (p, x + x', q', M \otimes M')$ if $q = p'$ and $\perp$ otherwise.

Semigroup of paths of $\mathcal{A}$ and $\mathcal{B}$. Recall we have fixed a deterministic max-plus automaton $\mathcal{A} = \langle Q_{\mathcal{A}}, \Sigma, M_{\mathcal{A}}, I_{\mathcal{A}}, F_{\mathcal{A}} \rangle$ and a max-plus automaton $\mathcal{B} = \langle Q_{\mathcal{B}}, \Sigma, M_{\mathcal{B}}, I_{\mathcal{B}}, F_{\mathcal{B}} \rangle$ over the same alphabet Σ.

Definition 4.3. The *semigroup of paths* of $\mathcal{A}$ and $\mathcal{B}$, denoted $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$, is the subsemigroup of $\overline{\mathfrak{M}_{Q_{\mathcal{A}},|Q_{\mathcal{B}}|}}$ generated by $\{(p, \overline{x}, q, \overline{M_{\mathcal{B}}(a)}) \mid a \in \Sigma, p \xrightarrow{a:x} q \text{ in } \mathcal{A}\}$.

Running Example, Part 4. From a and b respectively we construct

$$e_a = (p, 1, p, \begin{pmatrix} 0 & - & - & - \\ - & 1 & - & - \\ - & - & 0 & - \\ - & - & - & 0 \end{pmatrix}) \text{ and } e_b = (p, 1, p, \begin{pmatrix} 0 & 0 & - & - \\ - & - & 0 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}).$$

Observe that $e_a e_a = e_a$, and, for example, $e_a e_b$ and $e_b e_b$, in $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$, are given by

$$e_a e_b = (p, 1, p, \begin{pmatrix} 0 & 0 & - & - \\ - & - & 1 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}), e_b e_b = (p, 1, p, \begin{pmatrix} 0 & 0 & 0 & - \\ - & - & 0 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}).$$ ⌟

4.3. **Semigroup of asymptotic behaviours.**

The idempotent elements of a semigroup are elements e such that $e \otimes e = e$. A matrix M in $\mathcal{M}_{\Omega}^{i \times i}$ is called *path-idempotent* if $\overline{M}$ is idempotent in $\mathcal{M}_{\overline{\Omega}}^{i \times i}$. Similarly, an element (p, x, q, M) of $\mathfrak{M}_{Q,i}$ is called *path-idempotent* if $p = q$ and M is path-idempotent.

Stabilisation operation. The semigroup $\mathfrak{M}_{Q,i}$ is equipped with a unary operation on its path-idempotent elements, called the *stabilisation operation* and defined as follows: The stabilisation of elements in Ω is defined as: $(-\infty)^\sharp = -\infty$, $0^\sharp = 0$ and $1^\sharp = \infty^\sharp = \infty$. Given a path-idempotent matrix M in $\mathcal{M}_{\Omega}^{i \times i}$, the stabilisation of M, denoted $M^\sharp$ is defined as the product $M \otimes M' \otimes M$ where M' is the matrix M where all the diagonal elements are replaced by their stabilisation.

Definition 4.4. The *stabilisation operation* of $\mathfrak{M}_{Q,i}$ is defined on its path-idempotents as follows: $(p, x, p, M)^\sharp = (p, x^\sharp, p, M^\sharp)$.

Running Example, Part 5. Observe e_a is idempotent and

$$e_a^{\#} = (p, \infty, p, \begin{pmatrix} 0 & - & - & - \\ - & \infty & - & - \\ - & - & 0 & - \\ - & - & - & 0 \end{pmatrix})$$

indicating that the sequence of words $(a^n)_n$ has unbounded values in $\mathcal{A}$, and the sequence $(M_{\mathcal{B}}(a^n)_{q_2,q_2})_n$ is also unbounded.

Consider

$$e_a^{\#} e_b = (p, \infty, p, \begin{pmatrix} 0 & 0 & - & - \\ - & - & \infty & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}).$$

Recall from Running Example, Part 2 we are aiming to represent a word of the shape $(a^n b)^n$. Hence, we would expect to iterate $e_a^{\#} e_b$ again, however it is not path-idempotent. Instead we can make one iteration manually resulting in $e_a^{\#} e_b e_a^{\#} e_b$ which is idempotent,

$$e_a^{\#} e_b e_a^{\#} e_b = (p, \infty, p, \begin{pmatrix} 0 & 0 & \infty & - \\ - & - & \infty & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}).$$

Let us consider the effect of stabilisation on it. We have

$$(e_a^{\#} e_b e_a^{\#} e_b)^{\#} = (p, \infty, p, \begin{pmatrix} 0 & 0 & \infty & - \\ - & - & \infty & - \\ - & - & 0 & - \\ - & - & - & \infty \end{pmatrix}).$$

The purpose of this operation is to identify unbounded entries after arbitrarily many iterations of either the first, the second, or both, stabilisation operations. Note that $[\![\mathcal{A}]\!]((a^n b a^n b)^n) = 2n^2 + 2n$ while $M_{\mathcal{B}}((a^n b a^n b)^n)_{q_1,q_3} = M_{\mathcal{B}}((a^n b a^n b)^n)_{q_2,q_3} = n$ and $M_{\mathcal{B}}((a^n b a^n b)^n)_{q_4,q_4} = 2n$. Both sequences are unbounded as witnessed by the ∞ but this does not allow us to identify the different rates of growth between the entries. ⌟

Flattening operation. The semigroup $\mathfrak{M}_{Q,i}$ is also equipped with another unary operation on its path-idempotent elements, called the flattening operation and defined as follows: Given a path-idempotent matrix M in $\mathcal{M}_{\Omega}^{i\times i}$, the *flattening* of M, denoted $M^{\flat}$ is defined as the product $\overline{M} \otimes \langle M^3 \rangle \otimes \overline{M}$ where $\langle M \rangle$ is the matrix M where all the non diagonal elements are replaced by their barred version.

Definition 4.5. The *flattening operation* of $\mathfrak{M}_{Q,i}$ is defined on its path-idempotents as follows: $(p, x, p, M)^{\flat} = (p, x, p, M^{\flat})$.

Running Example, Part 6. Let us consider the effect of flattening on $e_a^{\#} e_b e_a^{\#} e_b$:

$$(e_a^{\#} e_b e_a^{\#} e_b)^{\flat} = (p, \infty, p, \begin{pmatrix} 0 & 0 & 1 & - \\ - & - & 1 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}).$$

Here observe that ∞ corresponds to an entry with growth rates n^2 on the sequence of words $(a^n b a^n b)^n$. Other entries, even unbounded sequences, with asymptotically smaller growth rates – such as n – are projected to 1 (or 0). The flattening behaviour allows us to capture differences in growth rates (not fully, but enough for our purpose), by, roughly speaking, keeping only the fastest growing elements. Intuitively $(e_a^{\#} e_b e_a^{\#} e_b)^{\flat}$ demonstrates that $\mathcal{A}$ can grow faster than $\mathcal{B}$ and so $\mathcal{A}$ is not big-O of $\mathcal{B}$, we formalise this intuition as a witness of non-domination in the next section. ⌟

The semigroup of asymptotic behaviours of $\mathcal{A}$ and $\mathcal{B}$. Recall we have fixed a deterministic max-plus automaton $\mathcal{A} = \langle Q_\mathcal{A}, \Sigma, M_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A}\rangle$ and a max-plus automaton $\mathcal{B} = \langle Q_\mathcal{B}, \Sigma, M_\mathcal{B}, I_\mathcal{B}, F_\mathcal{B}\rangle$ over the same alphabet Σ.

Definition 4.6. The *semigroup of asymptotic behaviours* of $\mathcal{A}$ and $\mathcal{B}$, denoted $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, is the subsemigroup of $\mathfrak{M}_{Q_\mathcal{A},|Q_\mathcal{B}|}$ generated by $\{(p, \overline{x}, q, \overline{M_\mathcal{B}(a)}) \mid a \in \Sigma, p \xrightarrow{a:x} q \text{ in } \mathcal{A}\}$ and closed under the stabilisation and flattening operations.

5. Decision Procedure

5.1. **Witnesses.** Given a deterministic max-plus automaton $\mathcal{A}$ and a max-plus automaton $\mathcal{B}$ over the same alphabet, an element (p, x, q, M) in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ is called a *witness of non-domination* if:

- p is initial in $\mathcal{A}$,
- q is final in $\mathcal{A}$,
- $x = \infty$,
- $\overline{I_\mathcal{B}} \otimes M \otimes \overline{F_\mathcal{B}} < \infty$.

Theorem 5.1. *Given two max-plus automata $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A}$ is deterministic and $[\![\mathcal{B}]\!] : \Sigma^* \to \mathbb{N}$, the two following assertions are equivalent:*

- *$\mathcal{A}$ is big-O of $\mathcal{B}$.*
- *There is no witness of non-domination in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$.*

We will not prove Theorem 5.1 directly, rather we will prove a refined version of the theorem incorporating tractable witnesses, which are defined next.

5.2. **Tractable witnesses.** A witness could be any element in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, found through arbitrary application of product, stabilisation and flattening, satisfying the conditions. We now consider a restricted form of witness in which we place a restriction on the sequence of operations to construct it.

Definition 5.2 (Tractable witness of non-domination)**.** We say an element g in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ is a *tractable witness of non-domination* if it is both a witness of non-domination and of the form

$$g = g_0(g_1(\dots(g_{k-2}(g_{k-1}(g_k)^{\#}g'_{k-1})^\flat g'_{k-2})^\flat \dots)^\flat g'_1)^\flat g'_0,$$

for $k \leq 3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ and $g_i, g'_i \in \overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}} \cup \{\mathsf{id}\}$, where id is an added identity element of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ such that $e \otimes \mathsf{id} = \mathsf{id} \otimes e = e$ for all $e \in \mathfrak{M}_{\mathcal{A},\mathcal{B}}$.

Running Example, Part 7. $(e_a^{\#} e_b e_a^{\#} e_b)^\flat$ is a witness but not a tractable witness of non-domination. However, $(e_a e_b e_a^{\#} e_b)^\flat$ will turn out to be a tractable witness. Intuitively, it represents the sequence of words $(aba^n b)^n$, which is almost the same as the sequence used to show $\mathcal{A}$ is not big-O of $\mathcal{B}$ in Running Example, Part 2. ⌟

We will now strengthen Theorem 5.1, in which we add the condition that there is a tractable witness; this allows us to limit our search to tractable witnesses.

Theorem 5.3. *Given two max-plus automata $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A}$ is deterministic and $[\![\mathcal{B}]\!] : \Sigma^* \to \mathbb{N}$, the following assertions are equivalent:*

- *$\mathcal{A}$ is big-O of $\mathcal{B}$.*
- *There is no witness of non-domination in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$.*
- *There is no tractable witness of non-domination in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$.*

The benefit of a tractable witness will be that we can identify the existence of one in PSPACE. In the next section we present the PSPACE algorithm to detect a tractable witness and then we will prove the equivalences of Theorem 5.3 to conclude the result.

5.3. **PSPACE algorithm.** We define a non-deterministic procedure to construct a tractable witness from middle out, that runs in polynomial space. Since NPSPACE = PSPACE (from Savitch's theorem), this will allow us to conclude.

Any element $g \in \overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$ can be constructed using at most $|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ product operations from the generators. Suppose $g = g_1 \otimes \cdots \otimes g_m$ such that g_i are generators and m is minimal, then we can assume that $g_1 \otimes \cdots \otimes g_i$ is different from $g_1 \otimes \cdots \otimes g_j$ for each $i \neq j, i, j \leq m$. Thus $m \leq |\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$.

The procedure is as follows:

- Non-deterministically choose $k \leq 3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$.
- Let g be a non-deterministically chosen idempotent element of $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$, constructed in at most $|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ steps.
- Update g to be the stabilisation $g^{\#}$.
- Repeating for $i = k$ to $i = 0$, we update g with $(g_i \ g \ g_i')^{\flat}$ for some $g_i, g_i' \in \overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}} \cup \{\mathsf{id}\}$ in the following way:
  - Update g by non-deterministically choosing a generator or id and multiply on the left of g. Repeat up to $|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ many times.
  - Update g by non-deterministically choosing a generator or id and multiply on the right of g. Repeat up to $|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ many times.
  - Except for $i = 0$, check that g is path-idempotent and update g to be the flattening of g.
- Check if g is a witness.

At any moment we are only storing one element g of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, plus the space needed for doing the product, iteration and stabilisation operations, the current iteration and number of iterations i and k, and how many elements we have multiplied (on the left or on the right) with g. This requires only polynomial space. This results in an NPSPACE algorithm, which is equivalent to a PSPACE algorithm.

5.4. **Proof of Theorem 5.3.** To prove Theorem 5.3, we are going to prove the following result - the notions of factorisation trees and faults will be introduced in due course.

Theorem 5.4. *Given two max-plus automata $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A}$ is deterministic and $[\![\mathcal{B}]\!] : \Sigma^* \to \mathbb{N}$, the following assertions are equivalent:*

(1) *$\mathcal{A}$ is not big-O of $\mathcal{B}$.*
(2) *There is a witness of non-domination in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$.*
(3) *There is a tractable witness of non-domination in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$.*
(4) *Some word has a factorisation tree, of height at most $3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$, with a fault.*

Since a tractable witness is a special case of a witness, it is clear that (3) implies (2). The remainder of the paper will prove the remaining implications.

In Section 6, we introduce the notions of factorisation trees and faults. In Section 7, we prove that (1) implies (4) - by proving its contrapositive. In Section 8, we prove that (4) implies (3). Sections 7 and 8 are independent of each other, but rely on Section 6. Finally, in Section 9, we prove that (2) implies (1). This later section can be read independently of the other ones.

6. Factorisation trees and faults

Recall we have fixed a deterministic max-plus automaton $\mathcal{A} = \langle Q_\mathcal{A}, \Sigma, M_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A}\rangle$ and a max-plus automaton $\mathcal{B} = \langle Q_\mathcal{B}, \Sigma, M_\mathcal{B}, I_\mathcal{B}, F_\mathcal{B}\rangle$ over the same alphabet Σ.

6.1. **Factorisation trees.** Let $w = w_1 \cdots w_k$ with $w_1, \dots, w_k \in \Sigma$ such that $[\![\mathcal{A}]\!](w) \neq -\infty$ and let:

$$p_0 \xrightarrow{w_1:x_1} p_1 \xrightarrow{w_2:x_2} p_2 \cdots p_{k-1} \xrightarrow{w_k:x_k} p_k$$

be its unique accepting path in $\mathcal{A}$. Let $\alpha_w(w_i)$ be the element of $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$ defined as $(p_{i-1}, \overline{x_i}, p_i, \overline{M_\mathcal{B}(w_i)})$.

A factorisation tree on w is a finite ordered tree in which every node ν in the tree is labelled by an element in $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$, denoted $\alpha(\nu)$, such that:

- there are k leaves labelled with $\alpha_w(w_1), \dots, \alpha_w(w_k)$,
- internal nodes have two or more children, and are labelled by the product of the labels of their children: a node ν with children $\nu_1, \dots \nu_m$ for some $m \geq 2$, is labelled with $\alpha(\nu) = \alpha(\nu_1) \otimes \alpha(\nu_2) \otimes \cdots \otimes \alpha(\nu_m)$. A node with two children is called a product node,
- if a node has at least three children then the children and the node are all labelled by the same idempotent element - such a node is called an idempotent node (in particular, $\alpha(\nu) = \alpha(\nu_i)$ for all $i \leq m$).

Note that for a word w such that $[\![\mathcal{A}]\!](w) \neq -\infty$, no node in a factorisation tree on w can be labelled by $\perp$. It is also clear that the labelling of the root of the subtree with leaves $\alpha_w(w_i), \dots, \alpha_w(w_j)$ for some $i < j$, corresponds to the element in the semigroup of paths witnessing the existence of 0 or positive weights paths in $\mathcal{A}$ and $\mathcal{B}$ on the word $w_i \cdots w_j$.

Theorem 6.1 (Simon's Factorisation Theorem [Sim90])**.** *There exists a positive integer H such that for all $w \in \Sigma^*$, there exist a factorisation tree on w of height at most H.*

Note that H does not depend on the word w, only on the size of $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$. Further we have that $H \leq 3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}| - 1$ due to the bound of [Kuf08], which is tighter than the $3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ bound of Colcombet [Col07], and the original bound of $9|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ by Simon.

6.2. **Contributors.** Let t be a factorisation tree on a word w, such that $[\![\mathcal{A}]\!](w) \neq -\infty$. For each node ν of the tree, we define its set of contributors C_ν as follows, in a top-down manner:

- if the root is labelled (p, x, q, M), the contributors of the root is the set of pairs (i, j) such that i is initial in $\mathcal{B}$, j is final in $\mathcal{B}$ and $M_{i,j} \neq -\infty$.
- if a node has a set of contributors C, and has two children labelled (p, x, q, M) and (q, x, r, P),
  - the set of contributors of the left child is:
    $\{(i, \ell) \mid \exists j : (i, j) \in C, P_{\ell,j} \neq -\infty, M_{i,\ell} \neq -\infty\}$,

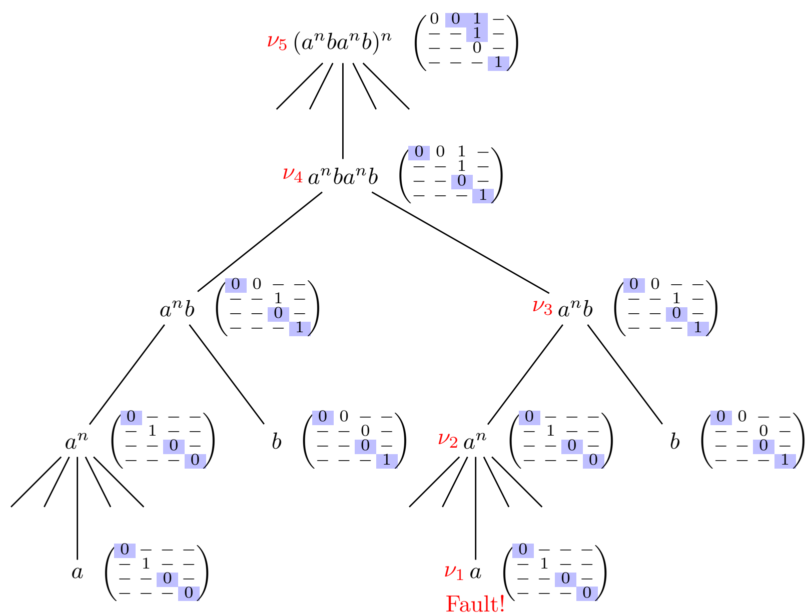


FIGURE 2. A possible factorisation tree for the word $w = (a^n b a^n b)^n$. Nodes are labelled by the sub-word of w and the path-behaviour of $M_{\mathcal{B}}$. The contributors of each node are indicated by highlighting the corresponding matrix entries. A node with a fault is also indicated, inducing the sequence of nodes $\nu_1, \ldots, \nu_5$ from the fault to the root.

  – the set of contributors of the right child is:
    $\{(\ell, j) \mid \exists i : (i, j) \in C, M_{i,\ell} \neq -\infty, P_{\ell,j} \neq -\infty\}$.
- if a node has a set of contributors C and has at least three children, labelled by an idempotent element (p, x, p, M), then:
  – the left-most child has set of contributors:
    $\{(i, \ell) \mid \exists j : (i, j) \in C, M_{\ell,j} \neq -\infty, M_{i,\ell} \neq -\infty\}$,
  – the right-most child has set of contributors:
    $\{(\ell, j) \mid \exists i : (i, j) \in C, M_{i,\ell} \neq -\infty, M_{\ell,j} \neq -\infty\}$,
  – the other children have set of contributors:
    $\{(\ell, k) \mid \exists (i, j) \in C, M_{i,\ell} \neq -\infty, M_{k,j} \neq -\infty,$
    $M_{\ell,k} \neq -\infty, M_{k,\ell} \neq -\infty\}$.

Contributors indicate which elements of the matrix meaningfully contribute to a valid run for the whole word. For example, at the root, not every entry is a weight on a run from an initial state to a final state, so not all entries contribute to the value computed on the word. The choices for the root and product nodes are uncontroversial, however the choice is non-trivial for the middle children of an idempotent node. Here only entries that could be repeated many times are taken; this is because, whilst other transitions could contribute to a valid run, they could contribute only once, whereas the entries we consider can be used in many of the idempotent children.

Running Example, Part 8. Fig. 2 depicts a factorisation tree for the word $(a^n b a^n b)^n$ of height 5. Every node is depicted with the sub-word and a partial description of its α-labelling: every node is α-labelled by $(p, 1, p, M)$, where M is the matrix depicted. The contributors are highlighted in the matrix for each node. Only a representation of the middle children is depicted for idempotent nodes. We will return to the fault label and the sequence $\nu_1, \ldots, \nu_5$ after defining faults. ⌟

We will make use of the property that every node has at least one contributor, stated in the following proposition.

Proposition 6.2. *Given a word w such that $[\![\mathcal{B}]\!](w) \neq -\infty$, every node in any factorisation tree on w has a non non-empty of contributors.*

Proof. We show that if a node has a contributor then all its children have a contributor. Since we assume $I_\mathcal{B} \otimes M_\mathcal{B}(w) \otimes F_\mathcal{B} \neq -\infty$ we observe that the root in any factorisation tree has non-empty contributors, this would imply that every element of the tree has non-empty contributors.

First observe, that the definition of a contributor requires that for every ν, with $\alpha(\nu) = (p, x, q, M)$, if $(i, j) \in C_\nu$ then $M_{i,j} \neq -\infty$. We consider two cases for both type of internal node. Suppose $(i, j) \in C_\nu$ and $M_{i,j} \neq -\infty$.

Case 1 (ν is a product node). Suppose $\alpha(\nu) = (p, x, q, M)$ is the product of two children ν_1 and ν_2 with $\alpha(\nu_1) = (p, x_1, r, P_1)$ and $\alpha(\nu_2) = (r, x_2, q, P_2)$ then $M = P_1 \otimes P_2$, and $M_{i,j} = (P_1)_{i,k} + (P_2)_{k,j}$ for some k. Thus $(i, k) \in C_{\nu_1}$ and $(k, j) \in C_{\nu_2}$.

Case 2 (ν is an idempotent node). Suppose ν has children $\nu_1, \ldots, \nu_k$ for $k \geq 3$, with $\alpha(\nu) = \alpha(\nu_i) = (p, x, p, M)$ for all $1 \leq i \leq k$. Since M is idempotent, we have $M = M^d$ for any choice of $d \geq 1$. Note that there is no requirement that $d \leq k$, the equivalence holds for *all* d by idempotence. In particular, let us fix a choice of some $d \geq |Q_\mathcal{B}| + 3$. Choose a contributor of ν, $(i, j) \in C_\nu$, and so we have $M_{i,j} \neq -\infty$. Further, since M is idempotent, we have $M_{i,j} = M^d_{i,j}$. Thus there exists a sequence $i = \ell_1, \ldots, \ell_{d+1} = j$ such that $M^d_{i,j} = M_{\ell_1,\ell_2} + M_{\ell_2,\ell_3} + \cdots + M_{\ell_d,\ell_{d+1}}$, where $M_{\ell_n,\ell_{n+1}} \neq -\infty$ for every $n \leq d$.

Note then that $(\ell_1, \ell_2) \in C_{\nu_1}, (\ell_d, \ell_{d+1}) \in C_{\nu_k}$. Furthermore, by simple application of the pigeon hole principle, there exists *distinct* indices n, m such that $i_n = i_m$ (as $d \geq |Q_\mathcal{B}| + 3$). Thus we have $(i_n, i_n) \in C_{\nu_2} \cap \cdots \cap C_{\nu_{k-1}}$. □

6.3. **Faults.** Given a word w such that $[\![\mathcal{A}]\!](w) \neq -\infty$ and a factorisation tree on w, a node labelled with (p, x, q, M), is called a *fault* if:

- it is the child of an idempotent node, but is neither the left-most nor the right-most child,
- $x = 1$,
- $M_{i,j} = 0$ for all pairs (i, j) in its set of contributors.

Running Example, Part 9. Let us return to the factorisation tree for $(a^n b a^n b)^n$ depicted in Fig. 2. Observe that the node indicated by ν_1 is a middle-child of an idempotent with only zero entries in the contributors and is therefore a fault. Since the two subtrees below ν_4 are identical the other node labelled by a is also a fault. In Running Example, Part 10 we will use the indicated fault ν_1 to construct a (tractable) witness of non-domination. ⌟

7. No tree has a fault implies Big-O

In this section, we suppose that no word has a factorisation tree of height at most $3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ with a fault, and we construct a positive integer c such that:

$$[\![\mathcal{A}]\!](w) \leq c[\![\mathcal{B}]\!](w) + c \quad \text{for all } w \in \Sigma^*. \tag{7.1}$$

Recall that we can also assume that $[\![\mathcal{B}]\!](w) \neq -\infty$ for all words w.

Let $w = w_1 \cdots w_k$ with $w_1, \ldots, w_k \in \Sigma$ such that $[\![\mathcal{A}]\!](w) \neq -\infty$ and let:

$$p_0 \xrightarrow{w_1:x_1} p_1 \xrightarrow{w_2:x_2} p_2 \cdots p_{k-1} \xrightarrow{w_k:x_k} p_k$$

be its unique accepting path in $\mathcal{A}$. Given a factorisation tree t on w, for a node ν in t, root of the subtree with leaves $\alpha_w(w_i), \ldots, \alpha_w(w_j)$, we denote by:

- $\mathsf{val}_{\mathcal{A}}(\nu)$ the weight $x_i + \ldots + x_j$ of the path in $\mathcal{A}$ corresponding to the factor $w_i \cdots w_j$,
- $\mathsf{Val}_{\mathcal{B}}(\nu)$ the matrix $M_{\mathcal{B}}(w_i \cdots w_j)$.

Let Λ be the largest value occurring on a transition in $\mathcal{A}$, and let $c_h = (4|Q_{\mathcal{B}}| + 4)^h \Lambda$ for positive integers h. We prove the following property:

Proposition 7.1. *Let w be a word such that $[\![\mathcal{A}]\!](w) \neq -\infty$ and t a factorisation tree on w with no fault. Let ν be a node in t of height h for some positive integer h. Then:*

$$\mathsf{val}_{\mathcal{A}}(\nu) \leq c_h \max_{(i,j) \in C_\nu} \mathsf{Val}_{\mathcal{B}}(\nu)_{i,j} + c_h$$

Observe that Eq. (7.1) is trivial for w such that $[\![\mathcal{A}]\!](w) = -\infty$. For w such that $[\![\mathcal{A}]\!](w) \neq -\infty$, Eq. (7.1) is a direct corollary of Proposition 7.1 choosing $c = c_H$ where $H = 3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$, ν as the root of a factorisation tree on w of height at most H, which exists by Theorem 6.1.

Proof. The proof is by induction on h.

Case 1 (If ν is a leaf and $h = 0$). By definition of Λ as the largest value occurring on a transition in $\mathcal{A}$, and by definition of c_0, we have:

$$\mathsf{val}_{\mathcal{A}}(\nu) \leq \Lambda \leq c_0 \leq c_0 \max_{(i,j) \in C_\nu} \mathsf{Val}_{\mathcal{B}}(\nu)_{i,j} + c_0.$$

The last inequality holds only if the set of contributors C_ν is not empty, which is the case by Proposition 6.2 since we assume $[\![\mathcal{B}]\!](w) \geq 0$ for all $w \in \Sigma^*$.

Case 2 (If ν is a product node). Let ν_1 and ν_2 be the two children of ν. Then $\mathsf{val}_{\mathcal{A}}(\nu) = \mathsf{val}_{\mathcal{A}}(\nu_1) + \mathsf{val}_{\mathcal{A}}(\nu_2)$. By induction, for each child $m \in \{1, 2\}$ we have:

$$\mathsf{val}_{\mathcal{A}}(\nu_m) \leq c_{h-1} \max_{(i,j) \in C_{\nu_m}} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{i,j} + c_{h-1}$$

Suppose $\mathsf{val}_{\mathcal{A}}(\nu_1) \geq \mathsf{val}_{\mathcal{A}}(\nu_2)$ (the case $\mathsf{val}_{\mathcal{A}}(\nu_2) > \mathsf{val}_{\mathcal{A}}(\nu_1)$ is symmetric). Then:

$$\begin{aligned} \mathsf{val}_{\mathcal{A}}(\nu_1) + \mathsf{val}_{\mathcal{A}}(\nu_2) &\leq 2\mathsf{val}_{\mathcal{A}}(\nu_1) \\ &\leq 2c_{h-1} \max_{(i,j) \in C_{\nu_1}} \mathsf{Val}_{\mathcal{B}}(\nu_1)_{i,j} + 2c_{h-1} \end{aligned}$$

Consider $(d, f) \in C_{\nu_1}$ for which this maximum is attained. Since (d, f) is a contributor of a left child then there exists g such that $\mathsf{Val}_{\mathcal{B}}(\nu_2)_{f,g} \neq -\infty$ and (d, g) is a contributor of ν.

We get:

$$\begin{aligned}
\mathsf{val}_{\mathcal{A}}(\nu) &\leq \mathsf{val}_{\mathcal{A}}(\nu_1) + \mathsf{val}_{\mathcal{A}}(\nu_2) \\
&\leq 2c_{h-1}\mathsf{Val}_{\mathcal{B}}(\nu_1)_{d,f} + 2c_{h-1} \\
&\leq 2c_{h-1}(\mathsf{Val}_{\mathcal{B}}(\nu_1)_{d,f} + \mathsf{Val}_{\mathcal{B}}(\nu_2)_{f,g}) + 2c_{h-1} \\
&\leq 2c_{h-1} \max_{(i,j)\in C_\nu} \mathsf{Val}_{\mathcal{B}}(\nu)_{i,j} + 2c_{h-1} \\
&\leq c_h \max_{(i,j)\in C_\nu} \mathsf{Val}_{\mathcal{B}}(\nu)_{i,j} + c_h && (\text{as } c_h > 2c_{h-1}.)
\end{aligned}$$

Case 3 (If ν is an idempotent node). Let $\nu_1, \ldots, \nu_d$ be the children of ν. By definition and idempotency, $\overline{\mathsf{val}_{\mathcal{A}}(\nu)} = \overline{\mathsf{val}_{\mathcal{A}}(\nu_m)}$ for all $m = 1, \ldots, d$. If $\overline{\mathsf{val}_{\mathcal{A}}(\nu)} = 0$, then we directly get the result since the set of contributors C_ν is not empty by Proposition 6.2 (because we assume that $[\![\mathcal{B}]\!](w) \geq 0$ for all $w \in \Sigma^*$). Let's suppose now that $\overline{\mathsf{val}_{\mathcal{A}}(\nu)} = \overline{\mathsf{val}_{\mathcal{A}}(\nu_m)} = 1$ for all $m = 1, \ldots, d$.

By inductive hypothesis, for all $m = 1, \ldots, d$, we have:

$$\mathsf{val}_{\mathcal{A}}(\nu_m) \leq c_{h-1} \max_{(i,j)\in C_{\nu_m}} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{i,j} + c_{h-1}.$$

Since there is no fault in the tree, then for all m, there is some $(i, j) \in C_{\nu_m}$ such that $\mathsf{Val}_{\mathcal{B}}(\nu_m)_{i,j} \geq 1$, and hence:

$$\mathsf{val}_{\mathcal{A}}(\nu_m) \leq 2c_{h-1} \max_{(i,j)\in C_{\nu_m}} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{i,j}. \tag{7.2}$$

Let (i_m, j_m) be a pair in C_{ν_m} on which this maximum is attained. Note that $\overline{\mathsf{Val}_{\mathcal{B}}(\nu_m)}$ are the same for all m as $\overline{\mathsf{Val}_{\mathcal{B}}(\nu)}$, and that this is an idempotent matrix. Let us define $i \sim j$ if and only if $\overline{\mathsf{Val}_{\mathcal{B}}(\nu)_{i,j}} = \overline{\mathsf{Val}_{\mathcal{B}}(\nu)_{j,i}} \neq -\infty$. By idempotency, this gives an equivalence relation, and we denote by $\mathcal{S}_1, \ldots, \mathcal{S}_z$ its equivalence classes. Note that z is bounded by the number of states of $\mathcal{B}$. We now partition the set $\{1, \ldots, d\}$ (the children of the node ν) as the union Φ of the sets $\{1\}$, $\{d\}$, $\Gamma_{\mathcal{S}_f,\text{even}}$ and $\Gamma_{\mathcal{S}_f,\text{odd}}$ for all $f = 1, \ldots, z$, where:

$$\Gamma_{\mathcal{S}_f,\text{even}} = \{m \in \{2, \ldots, d-1\} \mid i_m, j_m \in \mathcal{S}_f \text{ and } m \text{ is even}\}$$

and

$$\Gamma_{\mathcal{S}_f,\text{odd}} = \{m \in \{2, \ldots, d-1\} \mid i_m, j_m \in \mathcal{S}_f \text{ and } m \text{ is odd}\}.$$

This gives a partition of $\{1, \ldots, d\}$ since (i_m, j_m) is in the set of contributors of ν_m and then i_m and j_m are in the same $\mathcal{S}_f$ for some f. We partition this way into even and odd indices to be able to reconstruct a path: if we select for example the even indices in one of the $\mathcal{S}_f$, by definition, we can construct a path in the automaton taking the transitions corresponding to these indices. Note that the number of sets forming Φ is bounded by $2(|Q_{\mathcal{B}}| + 1)$, where we recall that $|Q_{\mathcal{B}}|$ is the number of states of $\mathcal{B}$.

For each Γ in Φ, let $x_\Gamma = \sum_{m\in\Gamma} \mathsf{val}_{\mathcal{A}}(\nu_m)$, and denote by Γ_{max} one for which this sum is maximal. Observe that:

$$\mathsf{val}_{\mathcal{A}}(\nu) \leq 2(|Q_{\mathcal{B}}| + 1)x_{\Gamma_{max}}.$$

Sub-case 3.1 ($\Gamma_{max} = \Gamma_{\mathcal{S}_f,\text{odd}}$ for some f). We have:

$$\begin{aligned}
\mathsf{val}_{\mathcal{A}}(\nu) &\le 2(|Q_{\mathcal{B}}|+1)x_{\Gamma_{max}} \\
&= 2(|Q_{\mathcal{B}}|+1) \sum_{m\in\Gamma_{\mathcal{S}_f,\text{odd}}} \mathsf{val}_{\mathcal{A}}(\nu_m) \\
&\le 2(|Q_{\mathcal{B}}|+1) \sum_{m\in\Gamma_{\mathcal{S}_f,\text{odd}}} (2c_{h-1}\mathsf{Val}_{\mathcal{B}}(\nu_m)_{i_m,j_m}) && \text{(by Eq. (7.2))} \\
&= 4c_{h-1}(|Q_{\mathcal{B}}|+1) \sum_{m\in\Gamma_{\mathcal{S}_f,\text{odd}}} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{i_m,j_m}.
\end{aligned}$$

By definition of contributors, and construction of $\Gamma_{\mathcal{S}_f,\text{odd}}$, there exists (i,j) in C_ν and $i=\ell_0,\ell_1,\ldots,\ell_d=j$ such that $i_m=\ell_{m-1}$ and $j_m=\ell_m$ for all m in $\Gamma_{\mathcal{S}_f,\text{odd}}$ and $\mathsf{Val}_{\mathcal{B}}(\nu_m)_{\ell_{m-1},\ell_m}\neq-\infty$ for all $m=1,\ldots,d$, hence

$$\sum_{m\in\Gamma_{\mathcal{S}_f,\text{odd}}} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{i_m,j_m} \le \sum_{m=1}^{d} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{\ell_{m-1},\ell_m}.$$

Since we also have:

$$\max_{(i,j)\in C_\nu} \mathsf{Val}_{\mathcal{B}}(\nu)_{i,j} = \max_{(i,j)\in C_\nu} \max_{i=\ell_0,\ell_1,\ldots,\ell_d=j} \left(\sum_{m=1}^{d} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{\ell_{m-1},\ell_m}\right)$$

we obtain:

$$\begin{aligned}
\mathsf{val}_{\mathcal{A}}(\nu) &\le 4c_{h-1}(|Q_{\mathcal{B}}|+1) \sum_{m\in\Gamma_{\mathcal{S}_f,\text{odd}}} \mathsf{Val}_{\mathcal{B}}(\nu_m)_{i_m,j_m} \\
&\le 4c_{h-1}(|Q_{\mathcal{B}}|+1) \max_{(i,j)\in C_\nu} \mathsf{Val}_{\mathcal{B}}(\nu)_{i,j}.
\end{aligned}$$

Sub-case 3.2 ($\Gamma_{max} = \Gamma_{\mathcal{S}_f,\text{even}}$ for some f). This is similar to the previous case.

Sub-case 3.3 ($\Gamma_{max} = \{1\}$). This is similar to the product case. We have:

$$\begin{aligned}
\mathsf{val}_{\mathcal{A}}(\nu) &\le 2(|Q_{\mathcal{B}}|+1)x_{\Gamma_{max}} \\
&= 2(|Q_{\mathcal{B}}|+1)\mathsf{val}_{\mathcal{A}}(\nu_1) \\
&\le 4(|Q_{\mathcal{B}}|+1)c_{h-1} \max_{(i,j)\in C_{\nu_1}} \mathsf{Val}_{\mathcal{B}}(\nu_1)_{i,j} \\
&\le 4(|Q_{\mathcal{B}}|+1)c_{h-1} \max_{(i,\ell)\in C_{\nu}} \mathsf{Val}_{\mathcal{B}}(\nu)_{i,\ell}
\end{aligned}$$

(by definition of contributors of the left-most child.)

Sub-case 3.4 ($\Gamma_{max} = \{d\}$). This is symmetric to the previous case. □

8. Construction of a witness if there is a tree with a fault

We suppose that some word has a factorisation tree of height at most $3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ with a fault. We are going to construct a tractable witness of non-domination in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$.

Let ν be a fault of maximal height in this tree. Let $\nu_1=\nu$ and let ν_{h+1} be the direct parent of ν_h for $h=2,\ldots,m$, where ν_m is the root node. Before giving the construction of the tractable witness formally, we will see how to construct it on our running example.

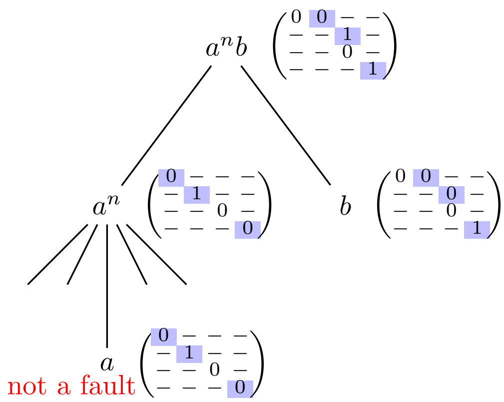


FIGURE 3. A factorisation tree for the word $a^n b$ with contributors highlighted.

Running Example, Part 10. We consider the tree depicted in Fig. 2. We define $\nu_1, \dots, \nu_5$ as explained above. A tractable witness is constructed by doing the stabilisation operation on the labelling of ν_1, then doing the product on the right with the labelling corresponding to b and on the left with the one corresponding to ab. This gets us to ν_4. At this point, we take the flattening of what we have obtained. We define the β-labelling of the nodes ν_h:

$$\begin{aligned}
\beta(\nu_2) = (p, 1, p, \begin{pmatrix} 0 & - & - & - \\ - & 1 & - & - \\ - & - & 0 & - \\ - & - & - & 0 \end{pmatrix})^\sharp &= (p, \infty, p, \begin{pmatrix} 0 & - & - & - \\ - & \infty & - & - \\ - & - & 0 & - \\ - & - & - & 0 \end{pmatrix}) \\
\beta(\nu_3) =& \beta(\nu_2) \otimes (p, 1, p, \begin{pmatrix} 0 & 0 & - & - \\ - & - & 0 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}) \\
=& (p, \infty, p, \begin{pmatrix} 0 & 0 & - & - \\ - & - & \infty & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}) \\
\beta(\nu_4) =& (p, 1, p, \begin{pmatrix} 0 & 0 & - & - \\ - & - & 1 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}) \otimes \beta(\nu_3) \\
=& (p, \infty, p, \begin{pmatrix} 0 & 0 & \infty & - \\ - & - & \infty & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix}) \\
\beta(\nu_5) = \beta(\nu_4)^\flat &= (p, \infty, p, \begin{pmatrix} 0 & 0 & 1 & - \\ - & - & 1 & - \\ - & - & 0 & - \\ - & - & - & 1 \end{pmatrix})
\end{aligned}$$

Observe that $\beta(\nu_5)$ is a witness of non-domination. ⌟

Running Example, Part 11. Recall that contributors are defined top down, thus whether a node is a fault depends on the context in which it sits. Observe that ν_5 of Fig. 2, which is a fault in that context, would not be fault if the tree were rooted at ν_3. This is because the contributors are different in this scenario, which is depicted in Fig. 3. In this case the node corresponding to a has a 1 in an entry of the contributors, and is therefore not a fault, and does not induce a witness. ⌟

Formally, every node ν in the tree is labelled by an element $\alpha(\nu)$ of $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$. We now associate nodes $\nu_2, \dots, \nu_m$ with elements of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, which we denote by $\beta(\nu)$, defined by the following:

- If $h = 2$, let $\beta(\nu_2) = \alpha(\nu_1)^{\#}$. This means we take stabilisation of the child's label.
- If ν_h is a product of ν_{h-1} and ν', we let $\beta(\nu_h) = \beta(\nu_{h-1}) \otimes \alpha(\nu')$.
- If ν_h is a product of ν' and ν_{h-1}, we let $\beta(\nu_h) = \alpha(\nu') \otimes \beta(\nu_{h-1})$.
- If ν_h is the idempotent product, with ν_{h-1} a middle child (that is, neither the left child nor the right) we let $\beta(\nu_h) = \beta(\nu_{h-1})^{\flat}$. This means we flatten the β-label of the child.
- If ν_h is the idempotent product, with ν_{h-1} as the left child (resp. right child), we let $\beta(\nu_h) = \beta(\nu_{h-1}) \otimes \alpha(\nu_{h-1})$ (resp. $\beta(\nu_h) = \alpha(\nu_{h-1}) \otimes \beta(\nu_{h-1})$).

Note we *only* associate β-labels with nodes on the path $\nu_2, \ldots, \nu_m$, and not other nodes in the tree.

We will observe that the β-labelling of the root, $\beta(\nu_m)$, is a tractable witness of non-domination. First, by construction, it has the correct shape: it is constructed with a single stabilisation at $\beta(\nu_2)$, and subsequently only products with elements of $\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$ and nested flattening operations. Since the height of the tree is bounded by $3|\overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}|$ then so to is the number of flattening operations defining $\beta(\nu_m)$. All is left is to show that it is a witness, which we do with the following property (in which we recall C_ν are the contributors of a node ν).

Proposition 8.1. *For all $h \in \{2, \ldots, m\}$, we have $\overline{\beta(\nu_h)} = \alpha(\nu_h)$ and if $\beta(\nu_h) = (p, x, q, M)$, we have $x = \infty$ and $M_{i,j} \leq 1$ for all $(i, j) \in C_{\nu_h}$.*

Observe that this immediately implies that $\beta(\nu_m) \in \mathfrak{M}_{\mathcal{A},\mathcal{B}}$ is a witness.

Proof. We proceed by induction on h starting from $h = 2$.

Case 1 (Base case, $h = 2$). Recall that $\beta(\nu_2) = \alpha(\nu)^{\#}$. Due to ν being an idempotent fault we can assume $\alpha(\nu)$ takes the form $(p, 1, p, M)$, hence $\beta(\nu_2) = (p, \infty, p, N)$ where $N = M \otimes M' \otimes M$, and M' replaces diagonal elements of M with their stabilisation, i.e. $M'_{\ell,\ell} = (M_{\ell,\ell})^{\#}$. Since ν is a fault $M_{\ell,\ell} = 0$ for $(\ell, \ell) \in C_\nu$, hence $M'_{\ell,\ell} = 0$. Thus for $(i, j) \in C_{\nu_2}$ $M_{i,j} = \max_{\ell,k} M_{i,\ell} + M'_{\ell,k} + M_{k,j} \leq 1$. Indeed for (ℓ, k) that could contribute to the maximum, $M_{i,\ell}, M_{k,j} \leq 1$ by definition, and either $\ell \neq k$ and $M'_{\ell,k} \leq 1$, or $\ell = k$, $(\ell, k) \in C_\nu$ and hence, $M'_{\ell,k} = 0$.

Case 2 (ν_h is a product). Let us assume ν_h is a product of ν_{h-1} and ν'. We then have $\overline{\beta(\nu_h)} = \overline{\beta(\nu_{h-1}) \otimes \alpha(\nu')} = \overline{\beta(\nu_{h-1})} \otimes \alpha(\nu') = \alpha(\nu_{h-1}) \otimes \alpha(\nu') = \alpha(\nu_h)$.

By induction hypothesis, $\beta(\nu_{h-1}) = (p, \infty, r, N')$ for some p, r, N'. Let us denote $\alpha(\nu') = (r, x, q, M)$ for some q, M and some finite x. Then $\beta(\nu_h) = (p, \infty, q, N' \otimes M)$. Let $N = N' \otimes M$.

Consider $(i, j) \in C_{\nu_h}$. We have $N_{i,j} = \max_\ell N'_{i,\ell} + M_{\ell,j}$. Observe that $(i, \ell) \in C_{\nu_{h-1}}$ whenever $M_{\ell,j} \neq -\infty$ and $N'_{i,\ell} \neq -\infty$. Hence $N'_{i,\ell} \leq 1$ by induction and $M_{\ell,j} \leq 1$ by definition, since $\alpha(\nu) \in \overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$. Hence $N_{i,j} \leq 1$.

The case ν_h is a product of some ν' and ν_{h-1} is similar.

Case 3 (ν_h is idempotent such that ν_{h-1} is the left-most child). First, note that we have $\alpha(\nu_h) = \alpha(\nu_{h-1}) = \overline{\beta(\nu_{h-1})}$ by induction, and $\overline{\beta(\nu_h)} = \overline{\beta(\nu_{h-1}) \otimes \alpha(\nu_h)}$ by definition. Hence, $\overline{\beta(\nu_h)} = \alpha(\nu_h)$ by idempotency. By induction hypothesis, $\beta(\nu_{h-1}) = (p, \infty, p, M)$, for some p, M, so $\beta(\nu_h) = (p, \infty, p, M \otimes \overline{M})$.

Let (i, j) in C_{ν_h}. Then by definition of contributors, for all ℓ such that $\overline{M}_{i,\ell} \neq -\infty$ and $\overline{M}_{\ell,j} \neq -\infty$, we have (i, ℓ) in $C_{\nu_{h-1}}$. Hence, by induction hypothesis, for such ℓ, $M_{i,\ell} \leq 1$. We then have $(M \otimes \overline{M})_{i,j} = \max_\ell M_{i,\ell} + \overline{M}_{\ell,j} \leq 1$.

Case 4 (ν_h is idempotent such that ν_{h-1} is the right-most child)*.* This case is symmetric to the previous one.

Case 5 (ν_h is idempotent such that ν_{h-1} is a middle child)*.* By idempotence, note that $\alpha(\nu_h) = \alpha(\nu_{h-1})$. Also note that for idempotent $e \in \overline{\mathfrak{M}}_{\mathcal{A},\mathcal{B}}$ we have $e^\flat = e$, thus $\overline{e^\flat} = \overline{e} = e$. Therefore we have $\overline{\beta(\nu_h)} = \overline{(\beta(\nu_{h-1}))^\flat} = \overline{\alpha(\nu_{h-1})^\flat} = \alpha(\nu_{h-1}) = \alpha(\nu_h)$.

By induction hypothesis, $\beta(\nu_{h-1}) = (p, \infty, p, M)$, for some M, and thus we have $\beta(\nu_h) = (p, \infty, p, M)^\flat = (p, \infty, p, M^\flat)$. Let $N = M^\flat$.

Consider $(i,j) \in C_{\nu_h}$. We have $N_{i,j} = \max_{\ell,k} \overline{M}_{i,\ell} + \langle M^3 \rangle_{\ell,k} + \overline{M}_{k,j}$, where $\langle M^3 \rangle$ is the matrix M^3 where all the non diagonal elements are replaced by their barred version. Observe that if $\ell \neq k$ then $\overline{M}_{i,\ell}, \langle M^3 \rangle_{\ell,k}, \overline{M}_{k,j}$ cannot be ∞, as each entry has been replaced by its barred version.

Thus it remains to verify $\overline{M}_{i,\ell} + \langle M^3 \rangle_{\ell,\ell} + \overline{M}_{\ell,j} \leq 1$. We show that $\langle M^3 \rangle_{\ell,\ell} \leq 1$ for ℓ such that $M_{i,\ell} \neq -\infty$ and $M_{\ell,j} \neq -\infty$. Note that $M^3_{\ell,\ell} \leq 1$ if and only if $M_{\ell,s} + M_{s,t} + M_{t,\ell} \leq 1$ for all s, t. For any such ℓ, s, t in which all three are not $-\infty$, the pairs $(\ell, s), (s, t)$ and (t, ℓ) are in the set of contributors $C_{\nu_{h-1}}$ of ν_{h-1}. Thus by induction all three are less than or equal to 1 and thus $M^3_{\ell,\ell} \leq 1$. □

9. Presence of witness implies non big-O

In this section, we assume that there is a witness of non-domination in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ and we construct a sequence of words $(w_i)_{i\in\mathbb{N}}$ such that for all positive integer c, there is i such that:

$$[\![\mathcal{A}]\!](w_i) > c \cdot [\![\mathcal{B}]\!](w_i) + c.$$

We will prove the following property:

Proposition 9.1. *For all (p, x, q, M) in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, for all $s \in \mathbb{N}$ there exists a pair (w_s, x_s), with w_s a word over Σ^* and $x_s \in \mathbb{N}$ with the following properties:*
(1) *$p \xrightarrow{w_s : x_s} q$ in $\mathcal{A}$ with $\overline{x} = \overline{x_s}$,*
(2) *$\overline{M_{\mathcal{B}}(w_s)} = \overline{M}$,*
(3) *if $x = \infty$, for all i, j such that $M_{i,j} \leq 1$, we have $x_s \geq s(M_{\mathcal{B}}(w_s))_{i,j} + s$.*

Note that applying this property to a witness of non-domination gives the expected result and concludes the proof.

Proof. We prove the proposition by structural induction on $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$. Consider an element (p, x, q, M) in $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$. By definition of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, (p, x, q, M) is either a generator of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ representing a letter, the product of two elements of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$, the stabilisation or the flattening of an element of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$.

Case 1 (Base case: generator representing letters)*.* Consider an element $(p, \overline{y}, q, \overline{M_{\mathcal{B}}(a)})$ such that $a \in \Sigma, p \xrightarrow{a:y} q$ in $\mathcal{A}$. We associate with every s the word $w_s = a$. Since $\overline{y} < \infty$, there is nothing to prove for (3).

Case 2 (Product of two elements)*.* Suppose $(p, x, q, M) = (p, y, r, N) \otimes (r, z, q, P)$ with $(u_s, y_s)_{s\in\mathbb{N}}$ and $(v_s, z_s)_{s\in\mathbb{N}}$ given by induction.

- If $y \leq 1$ and $z \leq 1$ (and so $x \leq 1$), we define $w_s = u_s v_s$ and $x_s = y_s + z_s$. (1) and (2) are immediate by definition and there is nothing to prove for (3).

- If $y = z = \infty$, we define $w_s = u_s v_s$ and $x_s = y_s + z_s$. (1) and (2) are immediate by definition. For (3), intuitively, since both $y = z = \infty$ we can straightforwardly bound both y_s and z_s through their respective words. Suppose $M_{i,j} \leq 1$, we have $M_\mathcal{B}(w_s)_{i,j} = M_\mathcal{B}(u_s)_{i,\ell} + M_\mathcal{B}(v_s)_{\ell,j}$ for some ℓ. Note that we have $N_{i,\ell} \leq 1$ and $P_{\ell,j} \leq 1$, otherwise $M_{i,j} = \infty$. Hence $s(M_\mathcal{B}(u_s)_{i,\ell}) + s \leq y_s$ and $s(M_\mathcal{B}(v_s)_{\ell,j}) + s \leq y_s$.

$$\begin{aligned} s(M_\mathcal{B}(w_s)_{i,j}) + s &= s(M_\mathcal{B}(u_s)_{i,\ell} + M_\mathcal{B}(v_s)_{\ell,j}) + s \\ &\leq sM_\mathcal{B}(u_s)_{i,\ell} + s + sM_\mathcal{B}(v_s)_{\ell,j} + s \\ &\leq y_s + z_s = x_s \quad \text{ as required.} \end{aligned}$$

- If $y = \infty$ but not z, let Θ be the maximum value appearing in the matrix $M_\mathcal{B}(v_0)$. We define $w_s = u_{s(\Theta+1)} v_0$ and $x_s = y_{s(\Theta+1)} + z_0$. (1) and (2) are immediate by definition. For (3), we will only be able to use property (3) inductively from $y = \infty$ but not z, thus we only use the short word v_0 (to ensure path compatibility) with the sufficiently larger word $u_{s(\Theta+1)}$. Suppose $M_{i,j} \leq 1$, we have $M_\mathcal{B}(w_s)_{i,j} = M_\mathcal{B}(u_{s(\Theta+1)})_{i,\ell} + M_\mathcal{B}(v_0)_{\ell,j}$ for some ℓ. Note that we have $N_{i,\ell} \leq 1$, otherwise $M_{i,j} = \infty$. Hence,

$$s(\Theta + 1)(M_\mathcal{B}(u_{s(\Theta+1)})_{i,\ell}) + s(\Theta + 1) \leq y_{s(\Theta+1)}.$$

$$\begin{aligned} s(M_\mathcal{B}(w_s)_{i,j}) + s &= s(M_\mathcal{B}(u_{s(\Theta+1)})_{i,\ell} + M_\mathcal{B}(v_0)_{\ell,j}) + s \\ &\leq sM_\mathcal{B}(u_{s(\Theta+1)})_{i,\ell} + s\Theta + s && \text{(since } M_\mathcal{B}(v_0)_{\ell,j} \leq \Theta) \\ &\leq s(\Theta + 1)M_\mathcal{B}(u_{s(\Theta+1)})_{i,\ell} + s(\Theta + 1) \\ &\leq y_{s(\Theta+1)} \leq y_{s(\Theta+1)} + z_0 = x_s. \end{aligned}$$

- The case of $z = \infty$ but not y is symmetric.

Case 3 (Stabilisation of an element)*.* Suppose $(p, x, p, M) = (p, y, p, P)^\sharp$ and $(u_s, y_s)_{s\in\mathbb{N}}$ given by induction.

- If $y = 0$ (and hence $x = 0$), let $w_s = u_s$ and $x_s = y_s$. (1) and (2) are immediate by definition - since $\overline{P^\sharp} = \overline{M}$ as P is path-idempotent - and there is nothing to prove for (3).
- If $y = \infty$ (and hence $x = \infty$), let $w_s = u_s$ and $x_s = y_s$. (1) and (2) are immediate by definition. Observe that if $M_{i,j} \leq 1$ then $P_{i,j} \leq 1$, thus

$$s(M_\mathcal{B}(w_s)_{i,j}) + s = s(M_\mathcal{B}(u_s)_{i,j}) + s \leq y_s = x_s.$$

- If $y = 1$ (and hence $x = \infty$), let Θ be the maximum value appearing in the matrix $M_\mathcal{B}(u_0)$ and recall $|Q_\mathcal{B}|$ is the number of states of $\mathcal{B}$. We define $w_s = u_0^{s(\Theta|Q_\mathcal{B}|+1)}$ and $x_s = s_0 s(\Theta|Q_\mathcal{B}| + 1)$. (1) and (2) are immediate by definition. For (3), intuitively if $M_{i,j} \leq 1$, then the repetition of u_0 cannot access a positive cycle between i and j, hence bounding the weight of $M_\mathcal{B}(w_s)_{i,j}$, while iterating u_0 sufficiently many times will make x_s as large as needed. Formally, if $M_{i,j} \leq 1$, then for all ℓ such that $P_{i,\ell}$ and $P_{\ell,j}$ are both different from $-\infty$, we have $P_{\ell,\ell} = 0$. Hence, since P is path-idempotent and $\overline{M_\mathcal{B}(u_0)} = \overline{P}$ by induction, $M_\mathcal{B}(w_s)_{i,j}$ has value at most $\Theta|Q_\mathcal{B}|$. On the other hand, the weight of w_s in $\mathcal{A}$ from p to p is at least $s(\Theta|Q_\mathcal{B}| + 1)$, since $y = 1$. Hence,

$$s(M_\mathcal{B}(w_s)_{i,j}) + s \leq s\Theta|Q_\mathcal{B}| + s \leq x_s.$$

Case 4 (Flattening of an element)*.* Suppose $(p, x, p, M) = (p, y, p, P)^\flat$ and $(u_s, y_s)_{s\in\mathbb{N}}$ given by induction.

- If $x = y \leq 1$ let $w_s = u_s$ and $x_s = y_s$. (1) and (2) are immediate by definition – since $\overline{P^\flat} = \overline{M}$ as P is path-idempotent – and there is nothing to prove for (3).
- Otherwise, we have $x = y = \infty$. Let Θ_s be the maximum value appearing in the matrix $M_{\mathcal{B}}(u_s)$, $|Q_{\mathcal{B}}|$ the number of states of $\mathcal{B}$ and $K_s = |Q_{\mathcal{B}}|\Theta_s + 1$. We define $w_s = (u_s)^{K_s}$ and $x_s = K_s y_s$. (1) and (2) are immediate by definition. We prove (3).

  By induction, if $P_{i,j} \leq 1$ then $s(M_{\mathcal{B}}(u_s)_{i,j}) + s \leq y_s$. Let $R_s = \max_{i,j : P_{i,j} \leq 1} M_{\mathcal{B}}(u_s)_{i,j}$. In particular,

$$sR_s + s \leq y_s. \tag{9.1}$$

  Consider i, j such that $M_{i,j} \leq 1$. Since P is path-idempotent and by definition of flattening, necessarily for all ℓ such that both $P_{i,\ell}$ and $P_{\ell,j}$ are different from $-\infty$, we have $P_{\ell,\ell} \leq 1$ $(\star)$.

  We have:

$$\begin{aligned} M_{\mathcal{B}}(w_s)_{i,j} &= M_{\mathcal{B}}(u_s^{K_s})_{i,j} \\ &= \max_{\substack{i_0,i_1,i_2,\ldots,i_{K_s} \\ i_0 = i,\ i_{K_s} = j}} M_{\mathcal{B}}(u_s)_{i_0,i_1} + M_{\mathcal{B}}(u_s)_{i_1,i_2} + \cdots + M_{\mathcal{B}}(u_s)_{i_{K_s-1},i_{K_s}} \end{aligned} \tag{9.2}$$

  Consider the path $i_0, i_1, \ldots, i_{K_s}$ that achieves the maximum in Eq. (9.2). Since we have $K_s \geq |Q_{\mathcal{B}}|$, there exists $n < m$ such that $i_n = i_m$ and all elements $i_0, \ldots, i_n, i_{m+1} \ldots, i_{K_s}$ are distinct. By $(\star)$, we have $P_{i_n,i_n} \leq 1$ and furthermore, $P_{i_\ell,i_{\ell+1}} \leq 1$ for $n \leq \ell \leq m-1$ since $P_{i_n,i_m} \leq 1$. By definition of R_s, we then have $M_{\mathcal{B}}(u_s)_{i_\ell,i_{\ell+1}} \leq R_s$. Hence all components of Eq. (9.2), *except* those between $i_0, \ldots, i_n$ and $i_m, \ldots, i_{K_s}$, are bounded above by R_s, and the remaining, of which there are at most $|Q_{\mathcal{B}}|$, are bounded above by Θ_s. We have:

$$M_{\mathcal{B}}(w_s)_{i,j} \leq |Q_{\mathcal{B}}|\Theta_s + K_s R_s. \tag{9.3}$$

  So, we have:

$$\begin{aligned} s(M_{\mathcal{B}}(w_s)_{i,j}) + s &\leq s(|Q_{\mathcal{B}}|\Theta_s + K_s R_s) + s && \text{(by Eq. (9.3))} \\ &= s(K_s R_s + |Q_{\mathcal{B}}|\Theta_s + 1) \\ &= s(K_s R_s + K_s) && \text{(by choice of } K_s\text{)} \\ &= K_s(sR_s + s) \\ &\leq K_s y_s && \text{(by Eq. (9.1))} \\ &= x_s. \end{aligned}$$

  □

Running Example, Part 12. We compute the sequence w_s for the nodes inducing the tractable witness in our example:

- The leaves, labelled by a and b, induce the sequences $w_s = a$ for all s, and $w_s = b$ for all s respectively.
- $\beta(\nu_2)$ is generated as the stabilisation of an element with word $u_s = a$ in which $y = 1$ (Case 3.3), hence $w_s = a^{s(\Theta|Q_{\mathcal{B}}|+1)} = a^{5s}$, where $\Theta = \max_{i,j} M_{\mathcal{B}}(a)_{i,j} = 1$ and $|Q_{\mathcal{B}}| = 4$.
- $\beta(\nu_3)$ is the product of elements with $u_s = a^{5s}$ and $v_s = b$, where $y = \infty$, but not z, and so we have $w_s = u_{s(\Theta+1)} v_0 = a^{10s} b$, where $\Theta = \max_{i,j} M_{\mathcal{B}}(b)_{i,j} = 1$.
- $\beta(\nu_4)$ is the product of elements with $u_s = ab$ and $v_s = a^{10s} b$, where $z = \infty$ but not y, and so we have $w_s = ab v_{s(\Theta+1)} = aba^{20s} b$, as $\Theta = \max_{i,j} M_{\mathcal{B}}(ab)_{i,j} = 1$.
- The tractable witness $\beta(\nu_5)$ is the flattening of an element with $u_s = aba^{20s} b$, where $y = \infty$, thus (by Case 4.2) $w_s = u_s^{|Q_{\mathcal{B}}|\Theta+1} = (aba^{20s} b)^{4 \cdot 20s+1}$, where $\Theta = \max_{i,j} M_{\mathcal{B}}(u_s)_{i,j} = 20s$.

Hence, our witness shows that for every s, $w_s = (aba^{20s}b)^{80s+1}$ is a contradiction to $[\![\mathcal{A}]\!] \leq s[\![\mathcal{B}]\!] + s$. Apart from the additional complexity introduced by the constants, the sequence matches our expectations from Running Example, Parts 2 and 7.

Indeed, $[\![\mathcal{A}]\!](w_s) = (20s+3)(80s+1) = 800s^2 + 260s + 3$ increases quadratically in s, while $[\![\mathcal{B}]\!](w_s) = 160s + 2$, maximised by counting b's, only increases linearly in s, and in particular:

$$800s^2 + 260s + 3 > s(160s + 2) + s \text{ for every } s \in \mathbb{N}.$$

⌟

10. Max-Plus automata with increasingly complex witnesses

We have defined tractable witnesses for two max-plus automata $\mathcal{A}$ and $\mathcal{B}$ as a means to decide whether $\mathcal{A}$ is big-O of $\mathcal{B}$. One of the specific characteristics of tractable witnesses is the number of nested $\flat$ operations appearing in the expression representing the witness. In this section, we construct a sequence of pairs of max-plus automata $(\mathcal{A}_n, \mathcal{B}_n)_{n\geq 1}$ with $\mathcal{A}_n$ not big-O of $\mathcal{B}_n$ such that (1) there is a tractable witness between $\mathcal{A}_n$ and $\mathcal{B}_n$ that has $n-1$ nested $\flat$ operations, but (2) there is no tractable witness with strictly less than $n-1$ nested $\flat$ operations. We go even further by defining the $\sharp\flat$-height of an element of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ (essentially the minimal number of nested $\sharp$ and $\flat$ operations required to produce this element from the basic elements corresponding to the letters and product) and we prove that for $\mathcal{A}_n$ and $\mathcal{B}_n$ there is no witness in $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$ that are of $\sharp\flat$-height less than n.

Recall that, for $\mathcal{A}$ deterministic, we have defined $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ as generated by the set

$$X_{\mathcal{A},\mathcal{B}} = \{(p, \overline{x}, q, \overline{M_{\mathcal{B}}(a)}) \mid a \in \Sigma, p \xrightarrow{a:x} q \text{ in } \mathcal{A}\}$$

and closed under product, stabilisation and flattening operations. We now define $\sharp\flat$-expressions, these describe how an element of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ is generated from $X_{\mathcal{A},\mathcal{B}}$.

Definition 10.1. The set of $\sharp\flat$-expressions over $X_{\mathcal{A},\mathcal{B}}$ is defined by induction as the minimal set such that:

- for all $x \in X_{\mathcal{A},\mathcal{B}}$, x is a $\sharp\flat$-expression,
- if s and t are $\sharp\flat$-expressions, then st is a $\sharp\flat$-expression,
- if s is a $\sharp\flat$-expression, then $s^\sharp$ is a $\sharp\flat$-expression,
- if s is a $\sharp\flat$-expression, then $s^\flat$ is a $\sharp\flat$-expression.

We map every $\sharp\flat$-expressions over $X_{\mathcal{A},\mathcal{B}}$ to the element in $\mathfrak{M}_{\mathcal{A},\mathcal{B}} \cup \{\perp\}$ corresponding to doing the product, $\sharp$ and $\flat$-operations as in the expression. Additionally, the expression will be mapped to $\perp$ if a $\sharp$ and $\flat$ operations are applied to elements which are not path-idempotent. Formally, we define π the following projection from the set of $\sharp\flat$-expressions over $X_{\mathcal{A},\mathcal{B}}$ to $\mathfrak{M}_{\mathcal{A},\mathcal{B}} \cup \{\perp\}$ by induction:

- if $x \in X_{\mathcal{A},\mathcal{B}}$, then $\pi(x) = x$,
- if s and t are $\sharp\flat$-expressions over X, then $\pi(st) = \pi(s) \otimes \pi(t)$,
- if s is a $\sharp\flat$-expression over X, then $\pi(s^\sharp) = \pi(s)^\sharp$ if $\pi(s)$ is path-idempotent, and $\perp$ otherwise,
- if s is a $\sharp\flat$-expression over X, then $\pi(s^\flat) = \pi(s)^\flat$ if $\pi(s)$ is path-idempotent, and $\perp$ otherwise.

Definition 10.2. The $\sharp\flat$-height of a $\sharp\flat$-expression, denoted $\sharp\flat$-height$(\cdot)$ is defined by induction as follows:

- for all $x \in X_{\mathcal{A},\mathcal{B}}$, $\sharp\flat$-height$(x) = 0$,

- if s and t are $\sharp\flat$-expressions, then $\sharp\flat\text{-height}(st) = \max(\sharp\flat\text{-height}(s), \sharp\flat\text{-height}(t))$,
- if s is a $\sharp\flat$-expression, then $\sharp\flat\text{-height}(s^\sharp) = 1 + \sharp\flat\text{-height}(s)$,
- if s is a $\sharp\flat$-expression, then $\sharp\flat\text{-height}(s^\flat) = 1 + \sharp\flat\text{-height}(s)$,

Definition 10.3. The $\sharp\flat$-height of an element z of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ is defined as

$$\sharp\flat\text{-height}(z) = \min\{\sharp\flat\text{-height}(s) \mid \pi(s) = z\}.$$

Consider a tractable witness of non-domination $g \in \mathfrak{M}_{\mathcal{A},\mathcal{B}}$; which can be described by a $\sharp\flat$-expressions of the form

$$g_0(g_1(\dots(g_{k-2}(g_{k-1}(g_k)^{\#}g'_{k-1})^{\flat}g'_{k-2})^{\flat}\dots)^{\flat}g'_1)^{\flat}g'_0.$$

The $\sharp\flat$-height of this expression is k, which shows that the $\sharp\flat$-height of g is **at most** k; but there may be another $\sharp\flat$-expression for g with lower height. The following proposition shows that allowing general witnesses does not help in terms of $\sharp\flat$-height and tractable witnesses have $\sharp\flat$-height as low as can be.

Proposition 10.4. *For all max-plus automata $\mathcal{A}$ and $\mathcal{B}$ (with $\mathcal{A}$ deterministic), if there exist a witness of $\sharp\flat$-height n for some $n \in \mathbb{N}$ then there exist a tractable witness of $\sharp\flat$-height at most n.*

Proof. We will say that an element of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ has a tractable witness shape if it can be written as:

$$g_0(g_1(\dots(g_{k-2}(g_{k-1}(g_k)^{\#}g'_{k-1})^{\flat}g'_{k-2})^{\flat}\dots)^{\flat}g'_1)^{\flat}g'_0,$$

for some $k \geq 0$, where all the g_i and g'_i are products of elements in $X_{\mathcal{A},\mathcal{B}}$.

We prove by induction that for all elements (p, ∞, q, M) of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ of $\sharp\flat$-height n, there is an element (p, ∞, q, M') with a tractable witness shape and $\sharp\flat$-height at most n such that $M' \leq M$ (in a component-wise meaning, where $-\infty < 0 < 1 < \infty$). Applying this to a witness leads to the expected result.

Let z be an element (p, ∞, q, M) of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ of $\sharp\flat$-height n.

- If z is in $X_{\mathcal{A},\mathcal{B}}$, then z has a tractable witness shape and is of $\sharp\flat$-height $n = 0$.
- If z is equal to z_1z_2, then $z_1 = (p, x_1, r, M_1)$, $z_2 = (r, x_2, q, M_2)$, z_1 and z_2 have $\sharp\flat$-height at most n, $M = M_1 \otimes M_2$, and one of x_1 or x_2 (or both) is equal to ∞. Let's assume $x_1 = \infty$, the other case is similar. By induction hypothesis on z_1, there is an element $z'_1 = (p, \infty, r, M'_1)$ with a tractable witness shape and $\sharp\flat$-height at most n such that $M'_1 \leq M_1$. We deduce that the element $z'_1\overline{z_2} = (p, \infty, q, M'_1 \otimes \overline{M_2})$ has a tractable witness shape of $\sharp\flat$-height at most n and $M'_1 \otimes \overline{M_2} \leq M$. In this context, by $\overline{z_2}$ we mean the element of $\mathfrak{M}_{\mathcal{A},\mathcal{B}}$ generated by $\sharp\flat$-expression for z_2 in which the stabilisation and flattening operations are omitted.
- If z is equal to $z_1^\sharp$, then if $z_1 = (p, \infty, q, M_1)$, we can directly apply the induction hypothesis to z_1 and immediately get an element satisfying the conditions (without needing to $\sharp$ it). If $z_1 = (p, 1, q, M_1)$, then $\overline{z_1}^\sharp = (p, \infty, q, \overline{M_1}^\sharp)$ satisfies the condition: it has a tractable witness shape of $\sharp\flat$-height 1, so at most n and $\overline{M_1}^\sharp \leq M_1^\sharp \leq M$.
- If z is equal to $z_1^\flat$, then $z_1 = (p, \infty, q, M_1)$ and has $\sharp\flat$-height at most $n - 1$. We can apply the induction hypothesis to z_1. There is an element $z'_1 = (p, \infty, q, M'_1)$ with a tractable witness shape of $\sharp\flat$-height at most $n - 1$ such that $M'_1 \leq M_1$. Hence, ${z'_1}^\flat = (p, \infty, q, {M'_1}^\flat)$ has a tractable witness shape of $\sharp\flat$-height at most n with ${M'_1}^\flat \leq {M_1}^\flat \leq M$. $\square$

Remark 10.5. Our main result, Theorem 5.4, asserts four equivalencies which are proven in a cycle, Proposition 10.4 directly proves that (2) implies (3) in Theorem 5.4.

Theorem 10.6. *There exists a sequence of pairs of max-plus automata* $(\mathcal{A}_n, \mathcal{B}_n)_{n\geq 1}$ *such that for all* $n \geq 1$*,*

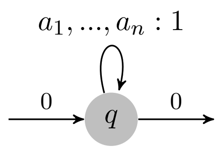
(1) *there exists a tractable witness in* $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$ *of* $\sharp\flat$*-height* n*,*
(2) *there is no witness in* $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$ *of* $\sharp\flat$*-height strictly less than* n*.*

In the rest of the paper, we fix a positive integer n. We give the construction of the max-plus automata $\mathcal{A}_n$ and $\mathcal{B}_n$ in Section 10.1, and prove items 1. in Section 10.2 and 2. in Section 10.3.

For the rest of the paper, $\Sigma_n = \{a_1, a_2, \dots, a_n\}$ denotes an alphabet with n letters[2].

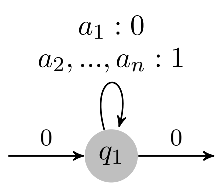
10.1. **Construction of $\mathcal{A}_n$ and $\mathcal{B}_n$.** The max-plus automata $\mathcal{A}_n$ and $\mathcal{B}_n$ are both defined on alphabet Σ_n. The automaton $\mathcal{A}_n$ is defined as the deterministic max-plus automata which computes the length of the input word, and we denote by q its only state.

$\mathcal{A}_n$:

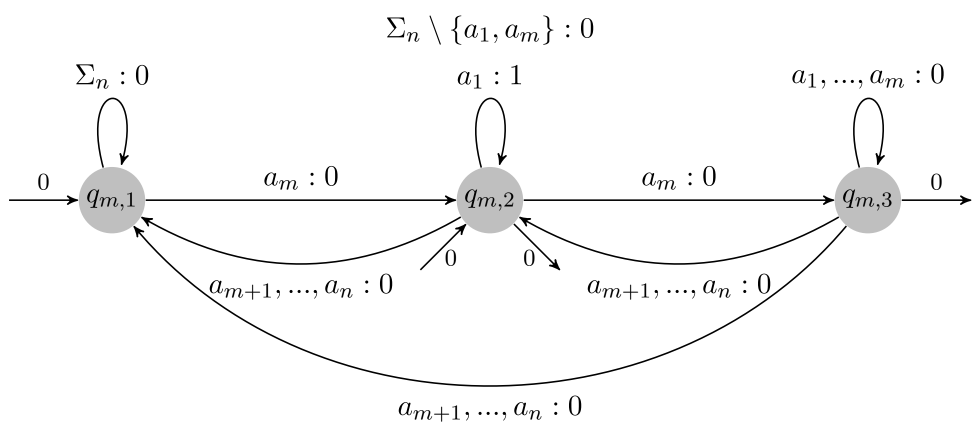


The automaton $\mathcal{B}_n$ is defined as the maximum (disjoint union) of n max-plus automata $\mathcal{B}_{n,1}, \mathcal{B}_{n,2}, \dots \mathcal{B}_{n,n}$. Automaton $\mathcal{B}_{n,1}$ has one state, and all other $\mathcal{B}_{n,m}$ have 3 states each (hence $\mathcal{B}_n$ has $3n - 2$ states).

$\mathcal{B}_{n,1}$:



For all $1 < m \leq n$, $\mathcal{B}_{n,m}$:



[2] The following construction would also work with a two letter alphabet, by decomposing each transition into $O(log(n))$ successive transitions in a standard way.

We define $\mathcal{B}_n$ as $\max(\mathcal{B}_{n,1}, \mathcal{B}_{n,2}, ..., \mathcal{B}_{n,n})$. By definition, the matrix associated to $\mathcal{B}_n$ for each word is a block diagonal matrix: the first block is of dimension 1×1 (for $\mathcal{B}_{n,1}$), and all other blocks are of dimension 3×3 (for $\mathcal{B}_{n,m}$, $2 \leq m \leq n$). Given a word w, we will denote by $M_1(w)$, $M_2(w)$,... $M_n(w)$ these blocks (matrices), omitting n which is now fixed.

Running Example, Part 13. Note that $\mathcal{A}_n, \mathcal{B}_n$ generalises the running example, that is, when $n = 2$, $\mathcal{A}_n, \mathcal{B}_n$ are exactly the $\mathcal{A}, \mathcal{B}$ from Running Example, Part 1. ⌟

Let us give a rough intuition of what automaton $\mathcal{B}_n$ is computing on words of the shape:

$$(a_n(a_{n-1}(\dots(a_2(a_1^{\ell_1})a_2)^{\ell_2}\dots a_{n-1})^{\ell_{n-1}}a_n)^{\ell_n}$$

- $\mathcal{B}_1 = \mathcal{B}_{1,1}$ computes the function that maps each word to 0.
- $\mathcal{B}_2 = \max(\mathcal{B}_{2,1}, \mathcal{B}_{2,2}) = \mathcal{B}$ from the running example (up to renaming of letters) and computes the function that maps each word to the maximum between the longest block of a_1's and the number of a_2's.
- $\mathcal{B}_3 = \max(\mathcal{B}_{3,1}, \mathcal{B}_{3,2}, \mathcal{B}_{3,3})$ computes a function that maps each word of the shape $(a_3(a_2(a_1^{\ell_1})a_2)^{\ell_2}a_3)^{\ell_3}$ to the maximum of the three following components:
  - (for $\mathcal{B}_{3,1}$) the number of a_2's and a_3's, so big-O of $\ell_2\ell_3$,
  - (for $\mathcal{B}_{3,2}$) between all pairs of consecutive a_3's, take the largest number of a_1's appearing between two a_2's, and then take the sum, so it would give big-O of $\ell_1\ell_3$,
  - (for $\mathcal{B}_{3,3}$) the largest number of a_1's appearing between two a_3's (ignoring a_2's), so big-O of $\ell_1\ell_2$.
- for the general case, $\mathcal{B}_n$ computes a function that maps each word of the shape

$$(a_n(a_{n-1}(\dots(a_2(a_1^{\ell_1})a_2)^{\ell_2}\dots a_{n-1})^{\ell_{n-1}}a_n)^{\ell_n}$$

  to the maximum of the n following components:
  - (for $\mathcal{B}_{n,1}$) the number of a_2's, a_3's,... and a_n's, so big-O of $\ell_2\ell_3\cdots\ell_n$,
  - (for $\mathcal{B}_{n,m}$ with $1 < m \leq n$) between all pairs of consecutive letters a_i's and a_j's with $i, j > m$, take the largest number of a_1's appearing between two a_m's (ignoring the other letters), and then take the sum, so it would give big-O of $\ell_1\ell_2\cdots\ell_{m-1}\ell_{m+1}\cdots\ell_n$.

10.2. **Existence of tractable witnesses of $\sharp\flat$-height n.** For $i = 1, \dots, n$, let $\boldsymbol{a_i}$ denote the element $(q, 1, q, \overline{M_{\mathcal{B}_n}(a_i)})$ corresponding to the letter a_i in $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$.

Proposition 10.7. *The element of* $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$

$$\boldsymbol{w} = (\boldsymbol{a_n}(\boldsymbol{a_{n-1}}(\dots(\boldsymbol{a_3}(\boldsymbol{a_2}(\boldsymbol{a_1})^{\#}\boldsymbol{a_2})^{\flat}\boldsymbol{a_3})^{\flat}\dots)^{\flat}\boldsymbol{a_{n-1}})^{\flat}\boldsymbol{a_n})^{\flat}$$

is a tractable witness of non-domination between $\mathcal{A}_n$ *and* $\mathcal{B}_n$ *of* $\sharp\flat$*-height* n.

Proof. First of all, it is immediate by definition that $\boldsymbol{w} = (q, \infty, q, M)$ for some M. We are left to prove that all the coefficients in M corresponding to a path from an initial state to a final state in $\mathcal{B}_n$ are not ∞. These coefficients are $(M_1)_{1,1}$, $(M_m)_{1,2}$, $(M_m)_{1,3}$, $(M_m)_{2,2}$ and $(M_m)_{2,3}$ for $2 \leq m \leq n$.

We have $M_1(a_1) = 0$, and hence $(M_1)_{1,1} \neq \infty$.

For $2 \leq m \leq n$ and $2 \leq i < m$, we have :

$$M_m(a_1) = \begin{pmatrix} 0 & - & - \\ - & 1 & - \\ - & - & 0 \end{pmatrix}, \; M_m(a_i) = \begin{pmatrix} 0 & - & - \\ - & 0 & - \\ - & - & 0 \end{pmatrix}, \text{ and } M_m(a_m) = \begin{pmatrix} 0 & 0 & - \\ - & - & 0 \\ - & - & 0 \end{pmatrix},$$

which gives (path-idempotency is satisfied at every step):

$$M_m(a_1)^\sharp = \begin{pmatrix} 0 & - & - \\ - & \infty & - \\ - & - & 0 \end{pmatrix} \text{ and } (M_m(a_{m-1})(...)^\flat M_m(a_{m-1}))^\flat = \begin{pmatrix} 0 & - & - \\ - & \infty & - \\ - & - & 0 \end{pmatrix}.$$

Finally,

$$M_m(a_m)(...)^\flat M_m(a_m) = \begin{pmatrix} 0 & 0 & \infty \\ - & - & 0 \\ - & - & 0 \end{pmatrix} \text{ and } (M(a_m)_m(...)^\flat M(a_m)_m)^\flat = \begin{pmatrix} 0 & 0 & 1 \\ - & - & 0 \\ - & - & 0 \end{pmatrix}.$$

Finally, using only products and $\flat$ operations (but no $\sharp$ operation) on matrices that do not contain any ∞ cannot create ∞. Hence,

$$(M(a_n)_m(\ldots(M(a_m)_m(...)^\flat M(a_m)_m)^\flat \ldots)^\flat M(a_n)_m)^\flat$$

does not contain any ∞. This is true for all $2 \leq m \leq n$, and thus all $(M_m)_{1,2}$, $(M_m)_{1,3}$, $(M_m)_{2,2}$ and $(M_m)_{2,3}$ are different from ∞, giving the expected result. □

Running Example, Part 14. Proposition 10.7 entails that $(e_b(e_a)^\sharp e_b)^\flat$ is also a witness of non-domination for $\mathcal{A}, \mathcal{B}$. ⌟

10.3. **Non-existence of witnesses of $\sharp\flat$-height strictly less than n.** Using Propositions 10.4, it is enough to prove that there is no tractable witness of $\sharp\flat$-height $n-1$ in $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$ to obtain the expected result. Note that, if there were a tractable witness of $\sharp\flat$-height $m < n-1$, then there would also exist a tractable witness of $\sharp\flat$-height $n-1$: for any matrix M, $(M^\sharp)^\flat \leq M^\sharp$ then by repeatedly using the $\flat$ operation after the (only) inner-most $\sharp$ operation we could get a tractable witness of any height higher than m.

Proposition 10.8. *There is no tractable witness of $\sharp\flat$-height $n-1$ in $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$.*

Let g be an element of $\mathfrak{M}_{\mathcal{A}_n,\mathcal{B}_n}$ with the shape of a tractable witness,

$$g = g_n(g_{n-1}(\ldots(g_2(g_1)^\# g'_2)^\flat \ldots)^\flat g'_{n-1})^\flat g'_n,$$

of $\sharp\flat$-height $n-1$, where each g_i and g'_i is a product (possibly empty, except for g_1) of some elements $\boldsymbol{a_j}$ for i ranging between 1 and n. We say that g_i has $\boldsymbol{a_j}$ as factor if it appears in it.

Let us first start with the intuition. In any tractable witness, g_1 must have $\boldsymbol{a_1}$ as its only factor, otherwise, $\mathcal{B}_{n,1}$ will keep up with $\mathcal{A}_n$ and it cannot be a witness. By using $\boldsymbol{a_1}$ in g_1, we have introduced ∞ into the matrix for every $\mathcal{B}_{n,m}$ ($m \neq 1$) and must remove this ∞ by a sequence of flattening operations. Essentially (although not exactly), the flattening operation removes ∞'s that are not on the diagonal. Therefore the g_i, g'_i ($i \geq 2$) in the witness must be used to take ∞'s off of the diagonal, so that flattening removes the ∞'s. However, the $\mathcal{B}_{n,m}$ are constructed in such a way that any g_i, g'_i that removes ∞'s in $\mathcal{B}_{n,m}$ cannot also remove ∞'s in any other $\mathcal{B}_{n,m'}$ ($m' \neq m$), resulting in the need for at least $n-1$ flattening operations; one for each $B_{n,m}$. We now prove the result formally, in which we must carefully keep track of the locations of each ∞.

Let $g = (q, x, q, M)$ for some x and M. We are going to prove that g cannot be a (tractable) witness. If $x \neq \infty$, the result is immediate. Otherwise, we denote by $M^{(1)}$, $M^{(2)}$,... $M^{(n)}$ the diagonal blocks (matrices) of M of size 1×1 for $M^{(1)}$ and 3×3 for

$M^{(m)}$, $m \neq 1$ corresponding to the behaviours in the automata $\mathcal{B}_{n,1}, \mathcal{B}_{n,2}, \dots \mathcal{B}_{n,n}$, and by $P_i^{(m)}, P_i'^{(m)}$ the matrices corresponding to the g_i, g_i' such that:

$$M^{(m)} = P_n^{(m)}(P_{n-1}^{(m)}(\dots(P_2^{(m)}(P_1^{(m)})^\sharp P_2'^{(m)})^\flat \dots)^\flat P_{n-1}'^{(m)})^\flat P_n'^{(m)}$$

Note that it is enough to prove that, either $M^{(1)} = (\infty)$ or there exist i between 2 and n such that one of $M_{1,2}^{(m)}$, $M_{2,2}^{(m)}$, $M_{1,3}^{(m)}$ or $M_{2,3}^{(m)}$ is ∞.

To prove this, we use the following three lemmas, their proofs follow in the following sections.

Lemma 10.9. *Let $1 < m \leq n$. If for all $1 \leq i < m$, $g_i g_i'$ has only factors $\boldsymbol{a_j}$ for $j \leq i$ then:*

$$(P_{m-1}^{(m)}(\dots(P_2^{(m)}(P_1^{(m)})^\sharp P_2'^{(m)})^\flat \dots)^\flat P_{m-1}'^{(m)})^\flat_{2,2} = \infty$$

Lemma 10.10. *Let $1 < m \leq n$. If for all $1 \leq i < m$, $g_i g_i'$ has only factors $\boldsymbol{a_j}$ for $j \leq i$ and $g_m g_m'$ has factor $\boldsymbol{a_j}$ for some $j > m$ then:*

$$(P_{n-1}^{(m)}(\dots(P_2^{(m)}(P_1^{(m)})^\sharp P_2'^{(m)})^\flat \dots)^\flat P_{n-1}'^{(m)})^\flat_{2,2} = \infty$$

Lemma 10.11. *Let $1 < m \leq n$. If P is a 3×3 matrix such that $P_{2,2} = \infty$, then the matrix $P_n^{(m)} P P_n'^{(m)}$ has ∞ in one of the $(1,2)$, $(2,2)$, $(1,3)$ or $(2,3)$ entries.*

Before proving these three lemmas, we show how they are used to prove that g is not a tractable witness, entailing Proposition 10.8.

Proof of Proposition 10.8. Let z be the smallest integer between 1 and n such that for all $1 \leq i < z$, $g_i g_i'$ has only factors $\boldsymbol{a_j}$ for $j \leq i$, and $g_z g_z'$ has factor $\boldsymbol{a_j}$ for some $j > z$; and $\perp$ if such an integer does not exist. We are in one of the following cases:

- $z = \perp$: In this case, for all $1 \leq i \leq n$, $g_i g_i'$ has only factors $\boldsymbol{a_j}$ for $j \leq i$. By denoting

  $$P = (P_{n-1}^{(n)}(\dots(P_2^{(n)}(P_1^{(n)})^\sharp P_2'^{(n)})^\flat \dots)^\flat P_{n-1}'^{(n)})^\flat$$

  we have, by Lemma 10.9 (applied to $m = n$),

  $$P_{2,2} = \infty$$

  and by Lemma 10.11,

  $$M^{(n)} = P_n^{(n)} P P_n'^{(n)}$$

  has ∞ in one of the $(1,2)$, $(2,2)$, $(1,3)$ or $(2,3)$ entries.
- $1 < z \leq n$: By denoting

  $$P = (P_{n-1}^{(z)}(\dots(P_2^{(z)}(P_1^{(z)})^\sharp P_2'^{(z)})^\flat \dots)^\flat P_{n-1}'^{(z)})^\flat$$

  we have, by Lemma 10.10 (applied to $m = z$),

  $$P_{2,2} = \infty$$

  and by Lemma 10.11,

  $$M^{(z)} = P_n^{(z)} P P_n'^{(z)}$$

  has ∞ in one of the $(1,2)$, $(2,2)$, $(1,3)$ or $(2,3)$ entries.
- $z = 1$: g_1 has factor $\boldsymbol{a_j}$ for some $j > 1$, and hence $M^{(1)} = (\infty)$. ∎

Proof of Lemmas 10.9 and 10.11.

Proof of Lemma 10.9. The proof is by induction on m. If $m = 2$, and g_1 has only factors $\boldsymbol{a_1}$ then:

$$(P_1^{(2)})^\sharp_{2,2} = \infty$$

Let $2 < m \leq n$, and suppose that for all $1 \leq i < m$, $g_i g_i'$ has only factors $\boldsymbol{a_j}$ for $j \leq i$. By induction, we have that

$$(P_{m-2}^{(m-1)}(\ldots(P_2^{(m-1)}(P_1^{(m-1)})^\sharp P_2'^{(m-1)})^\flat \ldots)^\flat P_{m-2}'^{(m-1)})^\flat_{2,2} = \infty$$

Since $\mathcal{B}_{n,m-1}$ and $\mathcal{B}_{n,m}$ are identical when restricted to letters $a_1, \ldots, a_{m-2}$, then:

$$(P_{m-2}^{(m)}(\ldots(P_2^{(m)}(P_1^{(m)})^\sharp P_2'^{(m)})^\flat \ldots)^\flat P_{m-2}'^{(m)})^\flat_{2,2} = \infty$$

Finally, since $g_{m-1}g_{m-1}'$ has only factors $\boldsymbol{a_j}$ for $j \leq m-1$:

$$(P_{m-1}^{(m)}(\ldots(P_2^{(m)}(P_1^{(m)})^\sharp P_2'^{(m)})^\flat \ldots)^\flat P_{m-1}'^{(m)})^\flat_{2,2} = \infty$$ □

Proof of Lemma 10.11. If g_n has factor $\boldsymbol{a_m}$, then $(P_n^{(m)})_{1,2} \neq -\infty$, otherwise $(P_n^{(m)})_{2,2} \neq -\infty$.

If g_n' does not have factor $\boldsymbol{a_m}$, or has at least one factor $\boldsymbol{a_m}$ that is later followed by a factor $\boldsymbol{a_j}$ for $j > m$, then $(P_n^{(m)})_{2,2} \neq -\infty$, otherwise, if no factor $\boldsymbol{a_m}$ is later followed by a factor $\boldsymbol{a_j}$ for $j > m$, $(P_n^{(m)})_{2,3} \neq -\infty$.

In all the cases, the matrix $P_n^{(m)} P P_n'^{(m)}$ has ∞ in one of the $(1,2)$, $(2,2)$, $(1,3)$ or $(2,3)$ entries. □

Proof of Lemma 10.10. Given a word w, we define $P_w^{(m)}$ as the 3×3 matrix of the behaviour of the automaton $\mathcal{B}_{n,m}$ on the word w. We extend this definition to $\sharp\flat$-expressions on Σ_n:

- If w is a word, then $P_w^{(m)}$ is defined above.
- If $w = w_1 w_2$ then $P_w^{(m)} = P_{w_1}^{(m)} P_{w_2}^{(m)}$.
- If $w = u^\sharp$ then $P_w^{(m)} = (P_u^{(m)})^\sharp$ provided $P_u^{(m)}$ is path-idempotent, otherwise it is undefined.
- If $w = u^\flat$ then $P_w^{(m)} = (P_u^{(m)})^\flat$ provided $P_u^{(m)}$ is path-idempotent, otherwise it is undefined.

Essentially, if $\pi(w) = (q, x, q, M)$, then $P_w^{(m)}$ is the restriction of M to matrix for $B_{n,m}$.

We say that a 3×3 matrix P satisfies condition (C) if the four following conditions are satisfied:

C1 : $P_{2,1} \neq -\infty$,
C2 : $P_{3,1} \neq -\infty$,
C3 : $P_{3,2} \neq -\infty$,
C4 : $P_{2,2} = \infty$.

Lemma 10.12. *Let v and w be two words and u a $\sharp\flat$-expression on Σ_n such that $(P_u^{(m)})$ is defined, $(P_u^{(m)})_{2,2} = \infty$ and vw has a letter a_j for some $j > m$ as factor. Then $(P_v^{(m)} P_u^{(m)} P_w^{(m)})^\flat$ satisfies (C).*

Lemma 10.13. *Let v and w be two words and u a $\sharp\flat$-expression on Σ_n such that $(P_u^{(m)})$ is defined and satisfies (C) then $(P_v^{(m)} P_u^{(m)} P_w^{(m)})^\flat$ satisfies (C).*

Let us first show how we can use Lemmas 10.12 and 10.13 to conclude the proof. We prove by induction that for all $m \leq k < n$,

$$(P_k^{(m)}(\dots(P_2^{(m)}(P_1^{(m)})^\sharp P_2'^{(m)})^\flat \dots)^\flat P_k'^{(m)})^\flat$$

satisfies (C). If so, by applying it to $k = n$ and considering (C4), this concludes the proof.

The base case ($k = m$) is given by Lemma 10.12 applied to

$$P_u^{(m)} = (P_{m-1}^{(m)}(\dots(P_2^{(m)}(P_1^{(m)})^\sharp P_2'^{(m)})^\flat \dots)^\flat P_{m-1}'^{(m)})^\flat$$

$P_v^{(m)} = P_m^{(m)}$ and $P_w^{(m)} = P_m'^{(m)}$. $(P_u^{(m)})_{2,2} = \infty$ by Lemma 10.9.

The induction case is given by Lemma 10.13.

Proof of Lemmas 10.12 and 10.13. First note the following properties:

Property 1: for all $\sharp\flat$-expressions w, $(P_w^{(m)})_{1,1} \neq -\infty$.

Property 2: for all $\sharp\flat$-expressions w, one (or both) of $(P_w^{(m)})_{2,2}$ or $(P_w^{(m)})_{2,3}$ is not $-\infty$.

Property 3: for all $\sharp\flat$-expressions w, one (or both) of $(P_w^{(m)})_{3,2}$ or $(P_w^{(m)})_{3,3}$ is not $-\infty$.

Property 4: for all words w that has a letter a_j for some $j > m$ as factor, $(P_w^{(m)})_{2,1}$ and $(P_w^{(m)})_{3,1}$ are not $-\infty$.

Property 5: for all words w that has a letter a_j for some $j > m$ as factor, $(P_w^{(m)})_{2,2}$ and $(P_w^{(m)})_{3,2}$ are not $-\infty$. (For entry $(2,2)$: Given a word w with a letter a_j for some $j > m$ as factor, (1) if w does not have letter a_m as factor then there is a path looping around $q_{m,2}$ on w, (2) if w has a letter a_m and all letters a_m are followed later in the word by a letter a_ℓ for some $\ell > m$ then loop around $q_{m,2}$ until reading a a_m, then go to $q_{m,3}$, loop around $q_{m,3}$ until seeing a letter a_ℓ for some $\ell > m$ and go back to $q_{m,2}$. Repeat, (3) if w has a letter a_m and some are not followed later by a letter a_ℓ for some $\ell > m$, then stay in $q_{m,2}$ or go to $q_{m,3}$ as long as no a_ℓ for some $\ell > m$ is seen. When the first one is read, go to $q_{m,1}$ and on the very last a_m go back to $q_{m,2}$. For entry $(3,2)$: Given a word w with a letter a_j for some $j > m$ as factor, (1) if w does not have letter a_m as factor then go to $q_{m,2}$ when reading the first a_j and loop in $q_{m,2}$ on the rest of the word, (2) if w has a letter a_m and all letters a_m are followed later in the word by a letter a_ℓ for some $\ell > m$ then loop around $q_{m,3}$ until reading a_ℓ for some $\ell > m$ and go to $q_{m,2}$, loop around $q_{m,2}$ until seeing a a_m, then go to $q_{m,3}$, loop around $q_{m,3}$ until seeing a letter a_ℓ for some $\ell > m$ and go back to $q_{m,2}$. Repeat, (3) if w has a letter a_m and some are not followed later by a letter a_ℓ for some $\ell > m$, then stay in $q_{m,3}$ or go to $q_{m,2}$ as long as no a_ℓ for some $\ell > m$ is seen. When the first one is read, go to $q_{m,1}$ and on the very last a_m go back to $q_{m,2}$.)

Property 6: for all $\sharp\flat$-expressions w, one (or both) of $(P_w^{(m)})_{1,2}$ or $(P_w^{(m)})_{2,2}$ is not $-\infty$.

Proof of Lemma 10.12. By path-idempotency, for (C1), (C2) and (C3), it is sufficient to prove that $P_v^{(m)} P_u^{(m)} P_w^{(m)}$ satisfies (C1), (C2) and (C3).

C1 ($P_{2,1} \neq -\infty$): If v has factor a_j for some $j > m$, then $(P_v^{(m)})_{2,1}$ is not $-\infty$ (Property 4). Additionally, $(P_u^{(m)})_{1,1}$ and $(P_w^{(m)})_{1,1}$ are not $-\infty$ (Property 1).

If w has factor a_j for some $j > m$, then neither $(P_w^{(m)})_{2,1}$ or $(P_w^{(m)})_{3,1}$ is $-\infty$ (Property 4). Additionally, $P_v^{(m)} P_u^{(m)} = P_{vu}^{(m)}$ and one of $(P_{vu}^{(m)})_{2,2}$ or $(P_{vu}^{(m)})_{2,3}$ is not $-\infty$

(Property 2).

In all cases, $(P_v^{(m)}P_u^{(m)}P_w^{(m)})_{2,1}$ is not $-\infty$.

C2 $(P_{3,1} \neq -\infty)$**:** If v has factor a_j for some $j > m$, then $(P_v^{(m)})_{3,1}$ is not $-\infty$ (Property 4). Additionally, $(P_u^{(m)})_{1,1}$ and $(P_w^{(m)})_{1,1}$ are not $-\infty$ (Property 1).

If w has factor a_j for some $j > m$, then neither $(P_w^{(m)})_{2,1}$ or $(P_w^{(m)})_{3,1}$ is $-\infty$ (Property 4). Additionally, $P_v^{(m)}P_u^{(m)} = P_{vu}^{(m)}$ and one of $(P_{vu}^{(m)})_{3,2}$ or $(P_{vu}^{(m)})_{3,3}$ is not $-\infty$ (Property 3).

In all cases, $(P_v^{(m)}P_u^{(m)}P_w^{(m)})_{3,1}$ is not $-\infty$.

C3 $(P_{3,2} \neq -\infty)$**:** If w has factor a_j for some $j > m$, then both $(P_w^{(m)})_{2,2}$ and $(P_w^{(m)})_{3,2}$ are not $-\infty$ (Property 5). Additionally, $P_v^{(m)}P_u^{(m)} = P_{vu}^{(m)}$ and one of $(P_{vu}^{(m)})_{3,2}$ or $(P_{vu}^{(m)})_{3,3}$ is not $-\infty$ (Property 3).

If v has factor a_j for some $j > m$, then both $(P_w^{(m)})_{3,1}$ and $(P_w^{(m)})_{3,2}$ are not $-\infty$ (Properties 4 and 5). Additionally, $P_u^{(m)}P_w^{(m)} = P_{uw}^{(m)}$ and one of $(P_{uw}^{(m)})_{1,2}$ or $(P_{uw}^{(m)})_{2,2}$ is not $-\infty$ (Property 6).

In all cases, $(P_v^{(m)}P_u^{(m)}P_w^{(m)})_{3,2}$ is not $-\infty$.

C4 $(P_{2,2} = \infty)$**:** We need to prove that:

$$\overline{P_v^{(m)}P_u^{(m)}P_w^{(m)}}\langle(P_v^{(m)}P_u^{(m)}P_w^{(m)})^3\rangle\overline{P_v^{(m)}P_u^{(m)}P_w^{(m)}}$$

satisfies (C4). First, $(\overline{P_v^{(m)}P_u^{(m)}P_w^{(m)}})_{2,2}$ is not $-\infty$ since vuw has a factor a_j for some $j > m$ (Property 5). Hence, it remains to prove that $((P_v^{(m)}P_u^{(m)}P_w^{(m)})^3)_{2,2}$ is ∞. We have one of $(P_v^{(m)}P_u^{(m)})_{2,2}$ or $(P_v^{(m)}P_u^{(m)})_{2,3}$ is not $-\infty$ (Property 2). Then both $(P_w^{(m)}P_v^{(m)})_{2,2}$ and $(P_w^{(m)}P_v^{(m)})_{3,2}$ are not $-\infty$ (Property 5). By hypothesis, we have $(P_u^{(m)})_{2,2} = \infty$. And finally, $(P_{wvuw}^{(m)})_{2,2}$ is not $-\infty$ (Property 5). Hence, $\langle(P_v^{(m)}P_u^{(m)}P_w^{(m)})^3\rangle_{2,2} = \infty$. □

Proof of Lemma 10.13. By path-idempotency, for (C1), (C2) and (C3), it is sufficient to prove that $P_v^{(m)}P_u^{(m)}P_w^{(m)}$ satisfies (C1), (C2) and (C3).

C1 $(P_{2,1} \neq -\infty)$**:** One of $(P_v^{(m)})_{2,2}$ or $(P_v^{(m)})_{2,3}$ is not $-\infty$ (Property 2). Additionally, $(P_u^{(m)})_{2,1}$ and $(P_u^{(m)})_{3,1}$ are not $-\infty$ by (C1) and (C2), and $(P_w^{(m)})_{1,1}$ is not $-\infty$ (Property 1). Hence $(P_v^{(m)}P_u^{(m)}P_w^{(m)})_{2,1}$ is not $-\infty$.

C2 $(P_{3,1} \neq -\infty)$**:** One of $(P_v^{(m)})_{3,2}$ or $(P_v^{(m)})_{3,3}$ is not $-\infty$ (Property 3). Additionally, $(P_u^{(m)})_{2,1}$ and $(P_u^{(m)})_{3,1}$ are not $-\infty$ by (C1) and (C2), and $(P_w^{(m)})_{1,1}$ is not $-\infty$ (Property 1). Hence $(P_v^{(m)}P_u^{(m)}P_w^{(m)})_{2,1}$ is not $-\infty$.

C3 $(P_{3,2} \neq -\infty)$**:** One of $(P_v^{(m)})_{3,2}$ or $(P_v^{(m)})_{3,3}$ is not $-\infty$ (Property 3). Additionally, $(P_u^{(m)})_{2,1}$, $(P_u^{(m)})_{2,2}$, $(P_u^{(m)})_{3,1}$ and $(P_u^{(m)})_{3,2}$ are not $-\infty$ by (C). Finally, one of $(P_w^{(m)})_{1,2}$ or $(P_w^{(m)})_{2,2}$ is not $-\infty$ (Property 6). Hence, overall, $(P_v^{(m)}P_u^{(m)}P_w^{(m)})_{3,2}$ is not $-\infty$.

C4 $(P_{2,2} = \infty)$**:** We need to prove that:

$$\overline{P_v^{(m)}P_u^{(m)}P_w^{(m)}}\langle(P_v^{(m)}P_u^{(m)}P_w^{(m)})^3\rangle\overline{P_v^{(m)}P_u^{(m)}P_w^{(m)}}$$

satisfies (C4). First, let's remark that for all words x and y, $(P_x^{(m)}P_u^{(m)}P_y^{(m)})_{2,2}$ is not $-\infty$. Indeed, one of $(P_x^{(m)})_{2,2}$ or $(P_x^{(m)})_{2,3}$ is not $-\infty$ (Property 2). Similarly, one of $(P_y^{(m)})_{1,2}$ or $(P_x^{(m)})_{2,2}$ is not $-\infty$ (Property 6). And additionally, all of $(P_u^{(m)})_{2,1}$, $(P_u^{(m)})_{2,2}$, $(P_u^{(m)})_{2,1}$, $(P_u^{(m)})_{3,2}$ are not $-\infty$ (condition (C)). Hence, $(P_x^{(m)}P_u^{(m)}P_y^{(m)})_{2,2}$ is not $-\infty$. That gives that: $(\overline{P_v^{(m)}P_u^{(m)}P_w^{(m)}})_{2,2}$ is not $-\infty$, leaving us to prove that $((P_v^{(m)}P_u^{(m)}P_w^{(m)})^3)_{2,2}$ is ∞. Still using the remark, we have that both $(P_v^{(m)}P_u^{(m)}P_{wv}^{(m)})_{2,2}$ and $(P_{wv}^{(m)}P_u^{(m)}P_w^{(m)})_{2,2}$ are not $-\infty$, which concludes the proof, using finally the hypothesis that $(P_u^{(m)})_{2,2} = \infty$. □

11. Conclusion

In this paper, we develop new techniques – in particular a new flattening operation – to describe the behaviour of max-plus automata. It would be interesting to see if such insight can be applied to other problems, particularly for min-plus automata. We also construct a series of examples of max-plus automata requiring tractable witnesses with an increasing number of nested flattening operations[3].

References

[ABK11] Shaull Almagor, Udi Boker, and Orna Kupferman. What's decidable about weighted automata? In *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011*, volume 6996 of *Lecture Notes in Computer Science*, pages 482–491. Springer, 2011. `doi:10.1007/978-3-642-24372-1_37`.

[ABK22] Shaull Almagor, Udi Boker, and Orna Kupferman. What's decidable about weighted automata? *Inf. Comput.*, 282:104651, 2022. Journal version of [ABK11]. `doi:10.1016/j.ic.2020.104651`.

[CD13] Thomas Colcombet and Laure Daviaud. Approximate comparison of distance automata. In *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013*, volume 20 of *LIPIcs*, pages 574–585. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. `doi:10.4230/LIPIcs.STACS.2013.574`.

[CDH10] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4):23:1–23:38, 2010. `doi:10.1145/1805950.1805953`.

[CDZ14] Thomas Colcombet, Laure Daviaud, and Florian Zuleger. Size-change abstraction and max-plus automata. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014*, volume 8634 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 2014. `doi:10.1007/978-3-662-44522-8_18`.

[CKMP20] Dmitry Chistikov, Stefan Kiefer, Andrzej S. Murawski, and David Purser. The big-O problem for labelled markov chains and weighted automata. In *31st International Conference on Concurrency Theory, CONCUR 2020*, volume 171 of *LIPIcs*, pages 41:1–41:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. Conference version of [CKMP22]. `doi:10.4230/LIPIcs.CONCUR.2020.41`.

[CKMP22] Dmitry Chistikov, Stefan Kiefer, Andrzej S. Murawski, and David Purser. The big-O problem. *Log. Methods Comput. Sci.*, 18(1), 2022. Journal version of [CKMP20]. `doi:10.46298/lmcs-18(1:40)2022`.

[CLM+22] Wojciech Czerwinski, Engel Lefaucheux, Filip Mazowiecki, David Purser, and Markus A. Whiteland. The boundedness and zero isolation problems for weighted automata over nonnegative rationals. In *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 15:1–15:13. ACM, 2022. `doi:10.1145/3531130.3533336`.

[3]We thank Ismaël Jecker for interesting discussions on this.

[Col07] Thomas Colcombet. Factorisation forests for infinite words. In *Fundamentals of Computation Theory, 16th International Symposium, FCT 2007*, volume 4639 of *Lecture Notes in Computer Science*, pages 226–237. Springer, 2007. doi:10.1007/978-3-540-74240-1_20.

[Col09] Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *Automata, Languages and Programming, 36th Internatilonal Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part II*, volume 5556 of *Lecture Notes in Computer Science*, pages 139–150. Springer, 2009. doi:10.1007/978-3-642-02930-1_12.

[DJL+21] Laure Daviaud, Marcin Jurdzinski, Ranko Lazic, Filip Mazowiecki, Guillermo A. Pérez, and James Worrell. When are emptiness and containment decidable for probabilistic automata? *J. Comput. Syst. Sci.*, 119:78–96, 2021. doi:10.1016/j.jcss.2021.01.006.

[DP23] Laure Daviaud and David Purser. The big-O problem for max-plus automata is decidable (PSPACE-complete). In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175798.

[FGKO15] Nathanaël Fijalkow, Hugo Gimbert, Edon Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Log. Methods Comput. Sci.*, 11(2), 2015. doi:10.2168/LMCS-11(2:12)2015.

[Has91] Kosaburo Hashiguchi. Algorithms for determining relative inclusion star height and inclusion star height. *Theor. Comput. Sci.*, 91(1):85–100, 1991. doi:10.1016/0304-3975(91)90269-8.

[KL09] Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPIcs*, pages 589–600. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009. doi:10.4230/LIPIcs.STACS.2009.1850.

[KMO+13] Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. On the complexity of equivalence and minimisation for Q-weighted automata. *Log. Methods Comput. Sci.*, 9(1), 2013. doi:10.2168/LMCS-9(1:8)2013.

[Kro94] Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *Int. J. Algebra Comput.*, 4(3):405–426, 1994. doi:10.1142/S0218196794000063.

[Kuf08] Manfred Kufleitner. The height of factorization forests. In *Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008*, volume 5162, pages 443–454. Springer, 2008. doi:10.1007/978-3-540-85238-4_36.

[Leu88] Hing Leung. On the topological structure of a finitely generated semigroup of matrices. In *Semigroup Forum*, volume 37, pages 273–287. Springer, 1988. doi:10.1007/BF02573140.

[MPR02] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Comput. Speech Lang.*, 16(1):69–88, 2002. doi:10.1006/csla.2001.0184.

[MS72] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory, College Park, Maryland, USA, October 25-27, 1972*, pages 125–129. IEEE Computer Society, 1972. doi:10.1109/SWAT.1972.29.

[Paz14] Azaria Paz. *Introduction to probabilistic automata*. Academic Press, 2014. doi:10.1016/C2013-0-11297-4.

[Sch61] Marcel Paul Schützenberger. On the definition of a family of automata. *Inf. Control.*, 4(2-3):245–270, 1961. doi:10.1016/S0019-9958(61)80020-X.

[Sim90] Imre Simon. Factorization forests of finite height. *Theor. Comput. Sci.*, 72(1):65–94, 1990. doi:10.1016/0304-3975(90)90047-L.

[Sim94] Imre Simon. On semigroups of matrices over the tropical semiring. *RAIRO Theor. Informatics Appl.*, 28(3-4):277–294, 1994. doi:10.1051/ita/1994283-402771.

[SM73] Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, 1973*, pages 1–9. ACM, 1973. doi:10.1145/800125.804029.

[Var85] Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 327–338. IEEE Computer Society, 1985. doi:10.1109/SFCS.1985.12.