THE BIG-O PROBLEM

DMITRY CHISTIKOV [a], STEFAN KIEFER [b], ANDRZEJ S. MURAWSKI [b], AND DAVID PURSER [c]

[a] Centre for Discrete Mathematics and its Applications (DIMAP) & Department of Computer Science, University of Warwick, UK

[b] Department of Computer Science, University of Oxford, UK

[c] Max Planck Institute for Software Systems, Saarland Informatics Campus, Germany

ABSTRACT. Given two weighted automata, we consider the problem of whether one is big-O of the other, i.e., if the weight of every finite word in the first is not greater than some constant multiple of the weight in the second.

We show that the problem is undecidable, even for the instantiation of weighted automata as labelled Markov chains. Moreover, even when it is known that one weighted automaton is big-O of another, the problem of finding or approximating the associated constant is also undecidable.

Our positive results show that the big-O problem is polynomial-time solvable for unambiguous automata, **coNP**-complete for unlabelled weighted automata (i.e., when the alphabet is a single character) and decidable, subject to Schanuel's conjecture, when the language is bounded (i.e., a subset of $w_1^* \ldots w_m^*$ for some finite words $w_1, \ldots, w_m$) or when the automaton has finite ambiguity.

On labelled Markov chains, the problem can be restated as a ratio total variation distance, which, instead of finding the maximum difference between the probabilities of any two events, finds the maximum ratio between the probabilities of any two events. The problem is related to ε-differential privacy, for which the optimal constant of the big-O notation is exactly $\exp(\varepsilon)$.

1. Introduction

Weighted automata over finite words are a well-known and powerful model of computation, a quantitative analogue of finite-state automata. Special cases of weighted automata include nondeterministic finite automata and labelled Markov chains, two standard formalisms for modelling systems and processes. Algorithms for analysis of weighted automata have been studied both in the early theory of computing and more recently by the infinite-state systems and algorithmic verification communities.

Given two weighted automata $\mathcal{A}$, $\mathcal{B}$ over an algebraic structure $(\mathcal{S}, +, \times)$, the equivalence problem asks whether the two associated functions $f_{\mathcal{A}}, f_{\mathcal{B}} \colon \Sigma^* \to \mathcal{S}$ are equal: $f_{\mathcal{A}}(w) = f_{\mathcal{B}}(w)$ for all finite words w over the alphabet Σ. Over the ring $(\mathbb{Q}, +, \times)$, equivalence is decidable in polynomial time by the results of Schützenberger [Sch61] and Tzeng [Tze92]; subsequently, fast parallel (**NC** and **RNC**) algorithms have been found for this problem [Tze96, KMO+13]. In contrast, for semirings the equivalence problem is hard: undecidable [Kro94, ABK11] for the semiring $(\mathbb{Q}, \max, +)$ and **PSPACE**-hard [MS72] for the Boolean semiring (for which weighted automata are usual nondeterministic finite automata and equivalence is equality of recognized languages). For the ring $(\mathbb{Q}, +, \times)$, replacing $=$ with $\leq$ makes the problem harder: the question of whether $f_{\mathcal{A}}(w) \leq f_{\mathcal{B}}(w)$ for all $w \in \Sigma^*$ is undecidable—even if $f_{\mathcal{A}}$ is constant [Paz14]. This problem subsumes the universality problem for (Rabin) probabilistic automata, yet another subclass of weighted automata (see, e.g., [Fij17]).

The problem of whether $f_{\mathcal{A}}(w) \leq f_{\mathcal{B}}(w)$ holds for all $w \in \Sigma^*$ has often been considered, under different semirings, but also for other forms of weighted automata, in which, e.g., $\limsup$ or (discounted) limit averages are used to combine the weights along a run. For example, most decidability and complexity results in [CDH10, BCV18] are on such weighted automata. Alternatively, to regain decidability in view of the undecidability result mentioned in the previous paragraph, one can consider semantic restrictions on the weighted automaton, e.g., the assumption that, for every w, all runs on w have the same weight. This route has been taken, e.g., in [FGR15].

In this paper, we introduce and study another natural problem, in which the condition $f_{\mathcal{A}}(w) \leq f_{\mathcal{B}}(w)$ is relaxed. Given $\mathcal{A}$ and $\mathcal{B}$ as above, is it true that there exists a constant $c > 0$ such that

$$f_{\mathcal{A}}(w) \leq c \cdot f_{\mathcal{B}}(w) \qquad \text{for all } w \in \Sigma^* ?$$

Using standard mathematical notation, this condition asserts that $f_{\mathcal{A}}(w) = O(f_{\mathcal{B}}(w))$ as $|w| \to \infty$, and we refer to this problem as the *big-O* problem accordingly.[1] The *big-Θ* problem (which turns out to be computationally equivalent to the big-O problem), in line with the $\Theta(\cdot)$ notation in analysis of algorithms, asks whether $f_{\mathcal{A}} = O(f_{\mathcal{B}})$ and $f_{\mathcal{B}} = O(f_{\mathcal{A}})$.

We restrict our attention to the ring $(\mathbb{Q}, +, \times)$ and only consider *non-negative weighted automata*, i.e., those in which all transitions have non-negative weights. We remark that, even under this restriction, weighted automata still form a superclass of (Rabin) probabilistic automata, a non-trivial and rich model of computation. Our initial motivation to study the big-O problem came from yet another formalism, labelled Markov chains (LMCs). One can think of the semantics of LMCs as giving a probability distribution or subdistribution on the set of all finite words. LMCs, often under the name Hidden Markov Models, are widely employed in a diverse range of applications; in computer-aided verification, they are perhaps the most fundamental model for probabilistic systems, with model-checking tools such as

---

[1]There also exists a related but slightly different definition of big-O; see Section 3.3 for details on the corresponding version of our big-O problem.

Prism [KNP11] or Storm [DJKV17] based on analyzing LMCs efficiently. All the results in our paper (including hardness results) hold for LMCs too. Our main findings are as follows.

- The big-O problem for non-negative WA and LMCs turns out to be **undecidable in general**, by a reduction from nonemptiness for probabilistic automata.
- In the **unary case**, i.e., if the input alphabet Σ is a singleton, the big-O problem becomes decidable and is, in fact, complete for the complexity class **coNP**. Unary LMCs are a simple and pure probabilistic model of computation: they run in discrete time and can terminate at any step; the big-O problem refers to this termination probability in two LMCs (or two WA). Our upper bound argument refines an analysis of growth of entries in powers of non-negative matrices by Friedland and Schneider [Sch86], and the lower bound is obtained by a reduction from unary NFA universality [SM73].
- In a more general **bounded case**, i.e., if the languages of all words w associated with non-zero weight are included in $w_1^* w_2^* \dots w_m^*$ for some finite words $w_1, \dots, w_m \in \Sigma^*$ (that is, are *bounded in the sense of Ginsburg and Spanier*; see [Gin66, Chapter 5] and [GS64]), the big-O problem is decidable subject to Schanuel's conjecture. This is a well-known conjecture in transcendental number theory [Lan66], which implies that the first-order theory of the real numbers with the exponential function is decidable [MW96]. Intuitively, our reliance on this conjecture is linked to the expressions for the growth rate in powers of non-negative matrices. These expressions are sums of terms of the form $\rho^n \cdot n^k$, where n is the length of a word, $k \in \mathbb{N}$, and ρ is an algebraic number. Our algorithms (however implicitly) need to compare for equality pairs of real numbers of the form $\log \rho_1 / \log \rho_2$, where ρ_i are algebraic, and it is an open problem in number theory whether there is an effective procedure for this task (the four exponentials conjecture asks whether two such ratios can ever be equal; see, e.g., Waldschmidt [Wal00, Sections 1.3 and 1.4]).

  Bounded languages form a well-known subclass of regular languages. In fact, a regular (or even context-free) language L is bounded if and only if the number of words of length n in L is at most polynomial in n. All other regular languages have, in contrast, exponential growth rate (a fact rediscovered multiple times; see, e.g., references in Gawrychowski et al. [GKRS10]). Bounded languages have been studied from combinatorial and algorithmic points of view since the 1960s [GS64, GKRS10], and have recently been used, e.g., in the analysis of quantitative information flow problems in computer security [Mes19b, Mes19a]. In the context of labelled Markov chains, languages that are subsets of $a_1^* a_2^* \dots a_m^*$ (for individual letters $a_1, \dots, a_m \in \Sigma$) model consecutive arrival of m events in a discrete-time system. It is curious that natural decision problems for such simple systems can lead to intricate algorithmic questions in number theory at the border of decidability.
- For **unambiguous automata**, i.e., where every word has at most one accepting path, the big-O problem is also decidable and can be solved in polynomial time.
- For **finitely ambiguous automata**, i.e., where there exists k, such that every word has at most k accepting paths, the big-O problem is decidable subject to Schanuel's conjecture, similarly to the case of bounded languages.

Further motivation and related work. In the labelled Markov chain setting, the big-O problem can be reformulated as a boundedness problem for the following function. For two LMCs $\mathcal{A}$ and $\mathcal{B}$, define the (asymmetric) *ratio variation function* by

$$r(\mathcal{A}, \mathcal{B}) = \sup_{E \subseteq \Sigma^*} \left(\frac{f_\mathcal{A}(E)}{f_\mathcal{B}(E)} \right),$$

where $f_{\mathcal{A}}(E)$ and $f_{\mathcal{B}}(E)$ denote the total probability mass associated with an arbitrary set of finite words $E \subseteq \Sigma^*$ in $\mathcal{A}$ and $\mathcal{B}$, respectively. Here we assume $\frac{0}{0} = 0$ and $\frac{x}{0} = \infty$ for $x > 0$. Observe that, because $\max(\frac{a}{b}, \frac{c}{d}) \geq \frac{a+c}{b+d}$ for $a, b, c, d \geq 0$, the supremum over $E \subseteq \Sigma^*$ can be replaced with supremum over $w \in \Sigma^*$. Consequently, the big-O problem for LMCs is equivalent to deciding whether $r(\mathcal{A}, \mathcal{B}) < \infty$, we give a formal proof of this in Section 3.

Finding the value of r amounts to asking for the optimal (minimal) constant in the big-O notation. Further, one can consider a symmetric variant, the *ratio distance*: $rd(\mathcal{A}, \mathcal{B}) = \max\{r(\mathcal{A}, \mathcal{B}), r(\mathcal{B}, \mathcal{A})\}$, by analogy with big-$\Theta$ (Proposition 3.2 also applies when r is replaced with rd). Now, rd is a ratio-oriented variant of the classic *total variation distance tv*, defined by $tv(\mathcal{A}, \mathcal{B}) = \sup_{E \subseteq \Sigma^*}(f_{\mathcal{A}}(E) - f_{\mathcal{B}}(E))$, which is a well-established way of comparing two labelled Markov chains [CK14, Kie18]. We also consider the problem of approximating r (as well as rd) to a given precision and the problem of comparing it with a given constant (threshold problem), showing that both are undecidable.

The ratio distance rd is also equivalent to the exponential of the *multiplicative total variation distance* defined in [CGPX14, Smi08] in the context of differential privacy. Consider a system $\mathcal{M}$, modelled by a single labelled Markov chain, where output words are observable to the environment but we want to protect the privacy of the starting configuration. Let $R \subseteq Q \times Q$ be a symmetric relation, which relates the starting configurations intended to remain indistinguishable. Given $\varepsilon \geq 0$, we say that $\mathcal{M}$ is ε-differentially private (with respect to R) if, for all $(s, s') \in R$, we have $f_s(E) \leq e^{\varepsilon} \cdot f_{s'}(E)$ for every (observable) set of traces $E \subseteq \Sigma^*$ [DMNS06, CMP19]. **Here in the subscript of f and elsewhere, references to states s and s' replace references to LMCs/automata: $\mathcal{M}$ stays implicit, and we specify which state it is executed from.**[2] Note that there exists such an ε if and only if $r(s, s') < \infty$ for all $(s, s') \in R$ or, equivalently, (the LMC $\mathcal{M}$ executed from) s is big-O of (the LMC $\mathcal{M}$ executed from) s' for all $(s, s') \in R$. In fact, the minimal such ε satisfies $e^{\varepsilon} = \max_{(s,s') \in R} r(s, s')$, thus r captures the level of differential privacy between s and s'.

Our results show that even deciding whether the multiplicative total variation distance is finite or $+\infty$ is, in general, impossible. Likewise, it is undecidable whether a system modelled by a labelled Markov chain provides any degree of differential privacy, however low.

2. Preliminaries

Let $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{R}$ be the natural, integer, rational and real numbers respectively. When accompanied by a constraint in the subscript, we restrict to the subset of numbers satisfying the constraint. For example, $\mathbb{N}_{\geq k}$ are the natural numbers greater than or equal to k, and in particular, $\mathbb{N}_{>0}$ are the positive natural numbers.

Definition 2.1. A *weighted automaton* $\mathcal{W}$ over the $(\mathbb{Q}, +, \times)$ semiring is defined as a 4-tuple $\langle Q, \Sigma, M, F \rangle$, where Q is a finite set of *states*, Σ is a finite *alphabet*, $M : \Sigma \to \mathbb{Q}^{Q \times Q}$ is a *transition weighting function*, and $F \subseteq Q$ is a set of *final states*. We consider only *non-negative* weighted automata, i.e. $M(a)(q, q') \geq 0$ for all $a \in \Sigma$ and $q, q' \in Q$.

In complexity-theoretic arguments, we assume that each weight is given as a pair of integers (numerator and denominator) in binary. The description size is then the number of bits required to represent $\langle Q, \Sigma, M, F \rangle$, including the bit size of the weights.

---

[2] It is standard that two automata $\mathcal{A}, \mathcal{B}$ can be merged into the same automaton by taking the disjoint union (relabeling states where necessary) and assigning s, s' as the starting state of $\mathcal{A}, \mathcal{B}$ respectively.

Each weighted automaton defines functions $f_s : \Sigma^* \to \mathbb{R}$, where for all $s \in Q$

$$f_s(w) = \sum_{t \in F} (M(a_1) \times M(a_2) \times \cdots \times M(a_n))_{s,t} \qquad \text{for } w = a_1 a_2 \ldots a_n \in \Sigma^*$$

and $A \times B$ is standard matrix multiplication. We refer to $f_s(w)$ as *the weight of w from state s*. Without loss of generality, a weighted automaton can have a single final state. If not, introduce a new unique final state t s.t. $M(a)(q, t) = \sum_{q' \in F} M(a)(q, q')$ for all $q \in Q$, $a \in \Sigma$.

We will typically define weighted automata by listing transitions as $q \xrightarrow[a]{p} q'$ (to mean $M(a)(q, q') = p$) with the assumption that any unspecified transition has weight 0.

Definition 2.2. We denote by $\mathcal{L}_s(\mathcal{W})$ the set of $w \in \Sigma^*$ with $f_s(w) > 0$, that is, with positive weight from s. Equivalently, this is the language of $\mathcal{N}_s(\mathcal{W})$, the non-deterministic finite automaton (NFA) formed from the same set of states (and final states) as $\mathcal{W}$, start state s, and transitions $q \xrightarrow{a} q'$ whenever $M(a)(q, q') > 0$.

Given $s, s' \in Q$, we say that s **is big-O of** s' if there exists $C > 0$ such that $f_s(w) \leq C \cdot f_{s'}(w)$ for all $w \in \Sigma^*$. The paper studies the following problem.

Definition 2.3 (Big-O Problem).
INPUT   Weighted automaton $\langle Q, \Sigma, M, F \rangle$ and $s, s' \in Q$
OUTPUT   Is s big-O of s'?

In the paper we also work with labelled Markov chains. In particular, they will appear in examples and hardness (including undecidability) arguments. As they are a special class of weighted automata, this will imply hardness (resp. undecidability) for weighted automata in general. On the other hand, our decidability results will be phrased using weighted automata, which makes them applicable to labelled Markov chains.

Definition 2.4. A *labelled Markov chain* (LMC) is a (non-negative) weighted automaton $\langle Q, \Sigma, M, F \rangle$ such that, for all $q \in Q \setminus F$, we have $\sum_{q' \in Q} \sum_{a \in \Sigma} M(a)(q, q') = 1$ and $M(a)(q, q') = 0$ for all $a \in \Sigma$, $q \in F$ and $q' \in Q$.

Since final states have no outgoing transitions, w.l.o.g., one can assume a unique final state. For LMCs, the function f_s can be extended to a measure on the powerset of Σ^* by $f_s(E) = \sum_{w \in E} f_s(w)$, where $E \subseteq \Sigma^*$. The measure is a subdistribution: $\sum_{w \in \Sigma^*} f_s(w) \leq 1$.

Probabilistic automata are similar to LMCs, except that $M(a)$ is stochastic for every a, rather than $\sum_{a \in \Sigma} M(a)$ being stochastic.

Definition 2.5. A *probabilistic automaton* $\mathcal{A}$ is a *non-negative weighted automaton* with a distinguished start state q_s such that $\sum_{q' \in Q} M(a)(q, q') = 1$ for all $q \in Q$ and $a \in \Sigma$. We use the notation $\mathbb{P}_{\mathcal{A}}(w) = f_{q_s}(w)$, where q_s is the start state of the probabilistic automaton $\mathcal{A}$.

We will also consider *unary* weighted automata, and similarly LMCs, where $|\Sigma| = 1$. Then we will often omit Σ on the understanding that $\Sigma = \{a\}$, and describe transitions with a single transition matrix $A = M(a)$ so that $f_s(a^n) = A^n_{s,t}$, where t is the unique final state. Note that $A^n_{s,t}$ stands for $(A^n)(s, t)$, and not $(A(s, t))^n$. Using the notation of regular expressions, we can write $\mathcal{L}_s(\mathcal{W}) \subseteq a^*$. It will turn out fruitful to consider several larger classes of languages:

Definition 2.6. Let $L \subseteq \Sigma^*$. L is *bounded* [GS64] if $L \subseteq w_1^* w_2^* \cdots w_m^*$ for some $w_1, \ldots, w_m \in \Sigma^*$. L is *letter-bounded* if $L \subseteq a_1^* a_2^* \ldots a_m^*$ for some $a_1, \ldots, a_m \in \Sigma$. L is *plus-letter-bounded* if $L \subseteq a_1^+ a_2^+ \ldots a_m^+$ for some $a_1, \ldots, a_m \in \Sigma$.

In each case, if the language of an NFA is suitably bounded, one can extract a corresponding bounding regular expression [GKRS10].

3. Related problems

In this section we show three related problems to the Big-O problem: the ratio variation function, the Big-Θ problem and the eventually Big-O problem. In particular, Observation 3.1, relating the ratio variation function, will be useful when we show undecidability.

3.1. **Ratio variation functions and distances.** We introduced in Section 1 the ratio variation function $r(s, s') = \sup_{E \subseteq \Sigma^*}(f_s(E)/f_{s'}(E))$, (now specified between two different starting states) and the ratio distance $rd(s, s') = \max\{r(s, s'), r(s', s)\}$ on labelled Markov chains. The supremum is taken over *sets* of words, reflecting its origin as a generalisation of the total variation distance. However, over labelled Markov chains this is not necessary, as the supremum function can be taken over individual words. The big-O problem is also taken over individual words, and taken in the formulation $r(s, s') = \sup_{w \in \Sigma^*}(f_s(w)/f_{s'}(w))$ the connection between the notion ratio variation function and the big-O problem becomes clear:

Observation 3.1. We have $r(s, s') < \infty$ if and only if s is big-O of s'. Furthermore, if finite, $r(s, s')$ is the minimum value of C witnessing that s is big-O of s'.

We now formally show this reformulation of the ratio variation distance.

Proposition 3.2. *For r on labelled Markov chains, it is sufficient to consider the supremum over $w \in \Sigma^*$ rather than $E \subseteq \Sigma^*$.*

Proof of Proposition 3.2. We will show we can approximate any event by a finite subset, then we can always simplify an event with more than one word, and not decrease the ratio.

Suppose $\frac{a+c}{b+d} > \frac{a}{b}$ and $\frac{a+c}{b+d} > \frac{c}{d}$. By the first inequality, we have $ab + bc > ab + ad$, so $bc > ad$. But by the second inequality, we have $ad + cd > bc + cd$, so $ad > bc$. This is a contradiction.

By repeated application of this technique, a *finite* set can always be simplified until the set has just a single element, resulting in a ratio which is no smaller. That is, there exists $w' \in E$ such that,

$$\frac{f_\mathcal{A}(E)}{f_\mathcal{B}(E)} = \frac{\sum_{w \in E} f_\mathcal{A}(w)}{\sum_{w \in E} f_\mathcal{B}(w)} \le \frac{f_\mathcal{A}(w')}{f_\mathcal{B}(w')}. \tag{3.1}$$

Now, we show that the consideration of finite sets is sufficient: consider an event $E \subseteq \Sigma^*$, then for every $\lambda > 0$ there is a $k \in \mathbb{N}$ such that $f_\mathcal{A}(E \cap \Sigma^{>k}) \le \lambda$, so then $f_\mathcal{A}(E \cap \Sigma^{\le k}) \le f_\mathcal{A}(E) \le f_\mathcal{A}(E \cap \Sigma^{\le k}) + \lambda$ [Kie18, Lemma 12]. For any ε, by choice of sufficiently small λ there is a finite set E' such that $\frac{f_\mathcal{A}(E')}{f_\mathcal{B}(E')} - \varepsilon \le \frac{f_\mathcal{A}(E)}{f_\mathcal{B}(E)} \le \frac{f_\mathcal{A}(E')}{f_\mathcal{B}(E')} + \varepsilon$.

Consider $\sup_{E \subseteq \Sigma^*} \frac{f_\mathcal{A}(E)}{f_\mathcal{B}(E)}$, this is equivalent to $\lim_{k \to \infty} \sup_{E \subseteq \Sigma^* \cap \Sigma^{\le k}} \frac{f_\mathcal{A}(E)}{f_\mathcal{B}(E)}$ and by Eq. (3.1) this is equivalent to $\lim_{k \to \infty} \sup_{w \in \Sigma^* \cap \Sigma^{\le k}} \frac{f_\mathcal{A}(w)}{f_\mathcal{B}(w)} = \sup_{w \in \Sigma^*} \frac{f_\mathcal{A}(w)}{f_\mathcal{B}(w)}$. □

(A) Reduction to big-Θ   (B) Reduction to big-O

FIGURE 1. Reductions between big-O and big-Θ

3.2. **The big-Θ problem.** One could consider whether s is big-Θ of s', defined as s is big-O of s' and s' is big-O of s; or equivalently for LMCs, whether $rd(s, s') < \infty$. We note that these two notions reduce to each other, justifying our consideration of only the big-O problem. There is an obvious reduction from big-Θ to big-O making two oracle calls (a Cook reduction), but, as we show below, this can be strengthened to a single call preserving the answer (a Karp reduction). In the other direction, one can ask if s big-O of s' using big-Θ by asking if a linear combination of s and s' is big-Θ of s': this ensures that the answer to s' is big-O of s and s' is always true, essentially leaving only s and s' big-O of s' to be checked, which depends only on whether s is big-O of s'.

Lemma 3.3. *The big-O problem is interreducible with the big-Θ problem.*

Proof.

Direction 1 (big-O problem reduces to the big-Θ problem). To ask if s is big-O of s', add states q, q' using the construction of Fig. 1a, then ask if q is big-Θ of q'.

$$\frac{f_q(aw)}{f_{q'}(aw)} = \frac{0.5f_s(w) + 0.5f_{s'}(w)}{f_{s'}(w)} < C \iff \frac{f_s(w)}{f_{s'}(w)} < 2C - 1$$

$$\frac{f_{q'}(aw)}{f_q(aw)} = \frac{f_{s'}(w)}{0.5f_s(w) + 0.5f_{s'}(w)} \le 2$$

Direction 2 (big-Θ problem reduces to the big-O problem). Given an automaton with weighting function f, let us construct a new automaton f' by replacing every transition $t = q \xrightarrow[a]{p} q'$ with $q \xrightarrow[a]{p} [t] \xrightarrow[a]{1} q'$, where $[t]$ is a new state. The effect of this procedure is that $f_s(a_1 \ldots a_n) = f'_s(a_1 a_1 \ldots a_n a_n)$. Any word of odd length has weight zero in f'. Given a word $w = a_1 \ldots a_n$, denote by $\tilde{w} = a_1 a_1 \ldots a_n a_n$.

Let us choose any character $a \in \Sigma$. We now add states $q, q', \bullet_1, \bullet_2$ as follows:

(1) $q \xrightarrow[a]{0.5} s$ and $q \xrightarrow[a]{0.5} \bullet_1 \xrightarrow[a]{1} s'$

(2) $q' \xrightarrow[a]{0.5} s'$ and $q' \xrightarrow[a]{0.5} \bullet_2 \xrightarrow[a]{1} s$

We claim that to ask if s is big-Θ of s' is equivalent to asking if q is big-O of q'. The idea of the construction is provided in Fig. 1b. Then we obtain the requisite bounds as follows:

$$\frac{f'_q(a\tilde{w})}{f'_{q'}(a\tilde{w})} = \frac{0.5f_s(w)}{0.5f_{s'}(w)} < C \iff \frac{f_s(w)}{f_{s'}(w)} < C,$$

$$\frac{f'_q(aa\tilde{w})}{f'_{q'}(aa\tilde{w})} = \frac{0.5f_{s'}(w)}{0.5f_s(w)} < C \iff \frac{f_{s'}(w)}{f_s(w)} < C.$$

Each of the reductions operates in logarithmic space.   □

3.3. **The Eventually Big-O Problem.** Readers familiar with the big-O notation may recall definition on $f, g : \mathbb{N} \to \mathbb{N}$, of the form: f is $O(g)$ if $\exists C, k > 0 \; \forall n > k \; f(n) \leq C \, g(n)$. Despite excluding finitely many points, when $g(n) \geq 1$, it is equivalent to the definition we use, $\exists C > 0 \; \forall n > 0 \; f(n) \leq C \, g(n)$, by taking C large enough to deal with the finite prefix.

In the paper, though, we formally consider s to not be big-O of s' if there exists even a single word w such that $f_s(w) > 0$ and $f_{s'}(w) = 0$. However, for weighted automata, we could amend our definition to "eventually big-O" as follows: $\exists C > 0, k > 0 : \forall w \in \Sigma^{\geq k} \; f_s(w) \leq C \cdot f_{s'}(w).$[3]

The Eventually Big-O Problem and Big-O Problem are also interreducible. We delay showing this equivalence to Section 6.1, as it depends on some further technical development.

Allowing additive error (in addition to the multiplicative error) in the inequality can also capture a similar relaxation of the Big-O problem. Colcombet and Daviaud [CD13] study the problem of *affine domination*, which asks whether there exists $c > 0$ such that $f_s(w) \leq c \cdot f_{s'}(w) + c$ for all $w \in \Sigma^*$? The problem is shown decidable for weighted automata over the tropical $(\mathbb{N} \cup \{\infty\}, \min, +)$ semiring, contrasting the containment problem $(f_s(w) \leq f_{s'}(w)$ for all $w \in \Sigma^*$?) which is also undecidable in that semiring. We observe that both containment and the big-O problem are undecidable in the $(\mathbb{Q}_{\geq 0}, +, \times)$ semiring that we are considering.

4. Big-O, Threshold and Approximation problems are undecidable

We show that the big-O problem is undecidable. We also establish undecidability for several other problems related to computing and approximating the ratio variation distance. Recall that this corresponds to identifying the optimal constant for positive instances of the big-O problem or the level of differential privacy between two states in a labelled Markov chain. Results in this section are presented on ratio total variation distances on *labelled Markov chains*, and thus apply to the big-O problem in the more general weighted automata.

In the following definition, a *promise problem* (see e.g., [ESY84]) restricts the inputs to the problem. An algorithm professing to decide the problem need only give a correct answer on the inputs conforming to the promise. If the input does not meet the promise, then the answer is not specified and can be arbitrary (including non-termination). In particular, it need not be decidable whether the promise holds.

Definition 4.1. The *asymmetric threshold problem* takes an LMC along with two states s, s' and a constant θ, and asks if $r(s, s') \leq \theta$. The variant under the promise of boundedness promises that $r(s, s') < \infty$; i.e., the input to the problem is restricted to the instances where it is known that $r(s, s') < \infty$.

The strict variant of each problem replaces $\leq$ with $<$.

The *asymmetric additive approximation task* takes an LMC, two states s, s' and a constant γ, and asks for x such that $|r(s, s') - x| \leq \gamma$. The *asymmetric multiplicative*

---

[3]It is instructive to juxtapose these definitions with classic definitions in analysis, following, e.g., de Bruijn's classic monograph [dB81, Section 1.2]. For two nonnegative functions $f, g : \mathbb{R} \to \mathbb{R}$:

— $f = O(g)$ on a set $S \subseteq \mathbb{R}$ if there is $c > 0$ such that $f(x) \leq c \cdot g(x)$ for all $x \in S$,

— $f = O(g)$ as $x \to a$ (as $x \to \infty$, respectively) if there is a neighbourhood of a (of ∞, respectively) such that $f = O(g)$ in that neighbourhood, in the sense defined above.

It is now clear that our definition of big-O from Section 2 asserts $f_s(w) = O(f_{s'}(w))$ for all $w \in \Sigma^*$, i.e., on the set Σ^*. In comparison, the "eventually big-O" asserts that $f_s(w) = O(f_{s'}(w))$ as $|w| \to \infty$.

approximation task takes an LMC, two states s, s' and a constant γ, and asks for x such that $1 - \gamma \leq \frac{x}{r(s,s')} \leq 1 + \gamma$.

In each case, the symmetric variant is obtained by replacing r with rd.

Theorem 4.2.
- *The big-O problem is undecidable, even for LMCs.*
- *Each variant of the threshold problem (asymmetric/symmetric, non-strict/strict) is undecidable, even under the promise of boundedness.*
- *All variants of the approximation tasks (asymmetric/symmetric, additive/multiplicative) are recursively unsolvable, even under the promise of boundedness.*

The proof of Theorem 4.2 will reduce from the emptiness problem for probabilistic automata (recall Definition 2.5). The problem EMPTY asks, given a probabilistic automaton $\mathcal{A}$, if $\mathbb{P}_\mathcal{A}(w) \leq \frac{1}{2}$ for all words $w \in \Sigma^*$. (Is the language $\{w \in \Sigma^* \mid \mathbb{P}_\mathcal{A}(w) > \frac{1}{2}\}$ empty?) It is known to be undecidable [Paz14, Fij17]. Recall, we use the notation $\mathbb{P}_\mathcal{A}(w)$ to refer to the probability mass in the probabilistic automaton, and the notation $f_s(w)$ for the weight (from state s) in the labelled Markov chain.

Proof idea for Theorem 4.2. We reduce from EMPTY. We assume we are given *any* probabilistic automaton $\mathcal{A}$, for which we would like to decide the emptiness problem. We first show a construction which will construct a specific labelled Markov chain from the probabilistic automaton. The resulting labelled Markov chain will have the property that answering any of the questions of Definition 4.1 would allow us to answer the emptiness problem for the probabilistic automaton $\mathcal{A}$. Therefore, an algorithm to decide any of the questions in Definition 4.1 would give an algorithm to decide the emptinenss problem. Since the emptiness problem is undecidabile, meaning there is no algorithm that can decide whether any given probabilistic automaton is empty or not, then there can be no algorithm to decide any of these questions.

The construction creates two branches of a labelled Markov chain. The first simulates the probabilistic automaton using the original weights multiplied by a scalar ($\frac{1}{4}$ in the case $|\Sigma| = 2$). The other branch will process each letter from Σ with equal weight (also $\frac{1}{4}$ in an infinite loop). Consequently, if there is a word accepted with probability greater than $\frac{1}{2}$, the ratio between the two branches will be greater than 1. The construction will enable words to be processed repeatedly, so that the ratio can then be pumped unboundedly. Certain linear combinations of the branches entail a gap promise, entailing undecidability of the threshold and approximation tasks. $\qquad\square$

We now implement the proof idea: first we show the properties required of the construction and (in Section 4.1) how these properties imply Theorem 4.2. Then we show the construction (in Section 4.2) and prove that the desired properties hold (in Section 4.3).

Given any probabilistic automaton $\mathcal{A}$, we construct a labelled Markov chain with three distinguished starting states s, s' and s''. These states will exhibit the following the properties:

Properties 4.3.
- If $\mathcal{A} \notin$ EMPTY then $r(s, s') = \infty$.
- If $\mathcal{A} \in$ EMPTY then $r(s, s') \leq 2$.
- If $\mathcal{A} \notin$ EMPTY then $49 < r(s, s'') \leq 51$.
- If $\mathcal{A} \in$ EMPTY then $r(s, s'') \leq 2$.
- $r(s', s) \leq 2$ and $r(s'', s) \leq 2$.

4.1. **Properties 4.3 imply Theorem 4.2.** The following lemma plays a key role in proving the result. In its statement, "undecidable to distinguish" means that the corresponding *promise problem* is undecidable. In other words, if the input is not in one of the two cases which should be distinguished between, the answer is not specified and can be arbitrary (including non-termination).

Due to the undecidability of Empty, a construction satisfying Properties 4.3 immediately establishes the following undecidability results:

Lemma 4.4.
- *Given an LMC along with two states s, s' and constant c, it is undecidable to distinguish between $r(s, s') \leq c$ and $r(s, s') = \infty$.*
- *Given an LMC along with two states s, s'' and two numbers c and C such that $c < C$, it is undecidable to distinguish between $r(s, s'') \leq c$ and $C \leq r(s, s'') < \infty$.*

Both statements remain true if r is replaced with rd. This is because $r(s', s) \leq 2$ and $r(s'', s) \leq 2$ ensures that $r(s, s')$ and $r(s, s'')$ dominate in the computation of rd, thus the proceeding four statements hold when replaced by rd.

Before we give the construction satisfying Properties 4.3, we show how Lemma 4.4 shows our theorem.

Proof of Theorem 4.2. We reason by contradiction using Lemma 4.4. For the big-O problem, it suffices to observe that, if it were decidable, one could use it to solve the first promise problem from the Lemma (recall that in a promise problem the input is guaranteed to fall into one of the two cases). This would contradict Lemma 4.4.

Similarly, the decidability of the (asymmetric) threshold problem would allow us to distinguish between $r(s, s') \leq c$ and $C \leq r(s, s') < \infty$ (second promise problem from the Lemma) by considering the instance $r(s, s') \leq \frac{c+C}{2}$ (non-strict variant) or $r(s, s') < \frac{c+C}{2}$ (strict variant). A positive answer (regardless of the variant) implies $r(s, s') \leq c$, while a negative one yields $r(s, s') \geq C$, which suffices to distinguish the cases. Note that in both cases $r(s, s')$ is bounded, so the reasoning remains valid if it is known in advance that $r(s, s')$ is bounded.

For additive (asymmetric) approximation, we observe that finding x such that $|r(s, s') - x| \leq \frac{C-c}{4}$ and comparing it with $\frac{c+C}{2}$ makes it possible to distinguish between $r(s, s') \leq c$ and $C \leq r(s, s') < \infty$. This is because $r(s, s') \leq c$ then implies $x < \frac{c+C}{2}$ and $C \leq r(s, s')$ implies $\frac{c+C}{2} < x$.

In the multiplicative case, finding x such that $1 - \frac{C-c}{4C} \leq \frac{x}{r(s,s')} \leq 1 + \frac{C-c}{4C}$ and comparing x with $\frac{c+C}{2}$ yields an analogous argument.

Since Lemma 4.4 also applies to rd, all of our results hold when r is replaced by rd. □

4.2. **The construction.** We now show how to build, given a probabilistic automaton, a labelled Markov chain that satisfies Properties 4.3. (We prove that the construction indeed satisfies these properties in the next section.)

Construction 4.5. For both cases, we reduce from Empty. We show our construction for $\Sigma = \{a, b\}$, but the procedure can be generalised to arbitrary alphabets.

The construction will create two branches of a labelled Markov chain. The first, from state q_s, will simulate the given probabilistic automaton using the original weights multiplied

by the same scalar (in this case $\frac{1}{4}$). The other branch, from state s_0, will process each letter from Σ with equal weight (also $\frac{1}{4}$ in an infinite loop). Consequently, if there is a word accepted with probability greater than $\frac{1}{2}$, the ratio between the two branches will be greater than 1. The construction will make it possible to process words repeatedly, so that the ratio can then be pumped unboundedly.

Formally, suppose we are given a probabilistic automaton $\mathcal{A} = \langle Q, \Sigma, M, F \rangle$ with start state q_s. First observe that w.l.o.g. q_s is not accepting, since in this case the empty word is accepted with probability 1, and thus there is a word with probability greater than $\frac{1}{2}$ and a trivial positive instance of the big-O problem can be returned.

We construct the LMC $\langle Q', \Sigma', \delta, F' \rangle$ taking $Q' = Q \uplus \{s, s', s'', s_0, t\}$ where $\uplus$ denotes disjoint union, $\Sigma' = \{a, b, acc, rej, \vdash\}$, $F' = \{t\}$ and δ as specified below. First we simulate the probabilistic automaton with a scaling factor of $\frac{1}{4}$: for all $q, q' \in Q$,

$$q \xrightarrow[a]{\frac{1}{4}M(a)(q,q')} q' \qquad q \xrightarrow[b]{\frac{1}{4}M(b)(q,q')} q'.$$

Originally accepting runs trigger a restart, while rejecting ones are redirected to t:

$$\text{if } q \in F : q \xrightarrow[acc]{\frac{1}{2}} q_s \quad \text{and} \quad \text{if } q \notin F : q \xrightarrow[rej]{\frac{1}{2}} t.$$

We then add a part of the chain which behaves equally, rather than according to the probabilistic automaton:

$$s_0 \xrightarrow[a]{\frac{1}{4}} s_0 \qquad s_0 \xrightarrow[b]{\frac{1}{4}} s_0 \qquad s_0 \xrightarrow[acc]{\frac{1}{4}} s_0 \qquad s_0 \xrightarrow[rej]{\frac{1}{4}} t.$$

The construction is illustrated in Fig. 2. To complete the reduction, we add the following transitions from s, s', s'':

$$s \xrightarrow[\vdash]{\frac{1}{2}} s_0 \qquad s \xrightarrow[\vdash]{\frac{1}{2}} q_s \qquad s' \xrightarrow[\vdash]{1} s_0 \qquad s'' \xrightarrow[\vdash]{\frac{99}{100}} s_0 \qquad s'' \xrightarrow[\vdash]{\frac{1}{100}} q_s.$$

4.3. Proof that the construction satisfies Properties 4.3.

It remains to show that Construction 4.5 satisfies Properties 4.3, which we show in the following two lemmas.

Lemma 4.6. *If $\mathcal{A} \notin$ EMPTY then $r(s, s') = \infty$. If $\mathcal{A} \in$ EMPTY then $r(s, s') \leq 2$. In either case, $r(s', s) \leq 2$.*

Proof. First observe that

$$\frac{f_s(\vdash w')}{f_{s'}(\vdash w')} = \frac{\frac{1}{2}f_{s_0}(w') + \frac{1}{2}f_{q_s}(w')}{f_{s_0}(w')} = \frac{1}{2} + \frac{1}{2}\frac{f_{q_s}(w')}{f_{s_0}(w')}. \tag{4.1}$$

If there is a word w that is accepted by the automaton with probability $> \frac{1}{2}$, then let $w'_i = (w\ acc)^i\ rej$ and we have

$$\frac{f_{q_s}(w'_i)}{f_{s_0}(w'_i)} = \frac{((\frac{1}{4})^{|w|}\mathbb{P}(w)\frac{1}{2})^i \cdot \frac{1}{2}}{((\frac{1}{4})^{|w|}\frac{1}{4})^i \cdot \frac{1}{4}} = 2 \cdot (2\mathbb{P}(w))^i. \tag{4.2}$$

Since $\mathbb{P}(w) > \frac{1}{2}$ then $2\mathbb{P}(w) > 1$ and we have:

$$\lim_{i \to \infty} \frac{f_s(\vdash w'_i)}{f_{s'}(\vdash w'_i)} = \infty \quad \text{and} \quad r(s, s') = rd(s, s') = \infty.$$
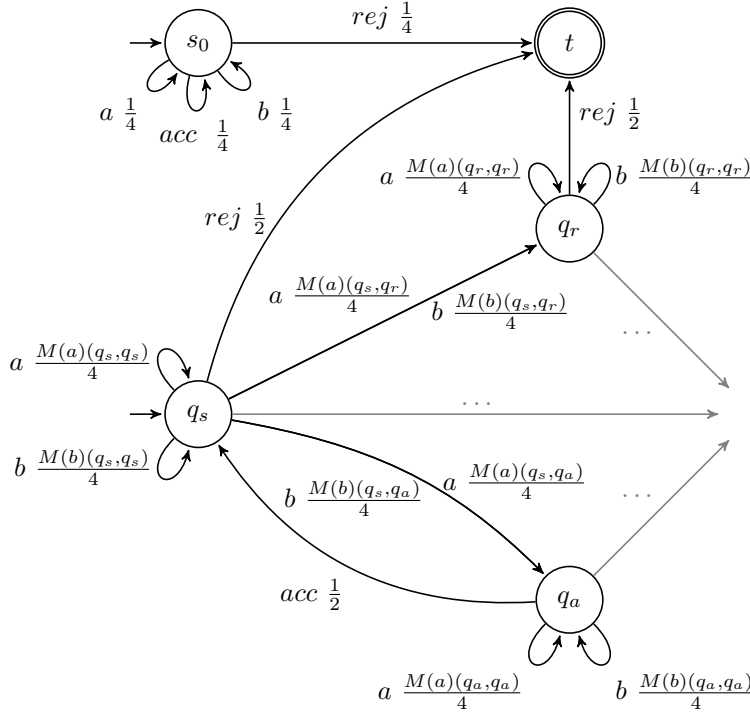
FIGURE 2. Reduction; where q_a represents accepting states of the probabilistic automaton, q_r represents rejecting states and q_s represents the start state (assumed to be rejecting).

If there is no such word then $\forall w \in \Sigma^* : \mathbb{P}(w) \leq \frac{1}{2}$, then probability ratio of all words is bounded. All words stating from states s and s' start with $\vdash$ and are terminated by rej, so in general all words take the form $w = \vdash (w_1\ acc) \dots (w_n\ acc)(w_{n+1}\ rej)$. Let us consider the probability of $w' = (w_1\ acc) \dots (w_n\ acc)(w_{n+1}\ rej)$ words from s_0 and q_s. Then:

$$\frac{f_{q_s}(w')}{f_{s_0}(w')} = \frac{\left(\prod_{i=1}^{n} \frac{1}{2}(\frac{1}{4})^{|w_i|}\mathbb{P}(w_i)\right)\left((\frac{1}{4})^{|w_{n+1}|}(1 - \mathbb{P}(w_{n+1}))\frac{1}{2}\right)}{(\frac{1}{4})^{|w_1|+\dots+|w_n|}(\frac{1}{4})^n(\frac{1}{4})^{|w_{n+1}|}\frac{1}{4}} \tag{4.3}$$

$$\leq \frac{\left((\frac{1}{4})^{|w_1|+\dots+|w_n|}(\frac{1}{2})^n(\frac{1}{2})^n\right)\left((\frac{1}{4})^{|w_{n+1}|}\frac{1}{2}\right)}{(\frac{1}{4})^{|w_1|+\dots+|w_n|+n}(\frac{1}{4})^{|w_{n+1}|}\frac{1}{4}} \qquad (\text{as } \forall i : \mathbb{P}(w_i) \leq \frac{1}{2})$$

$$= 2. \tag{4.4}$$

Then using Eq. (4.1) for every word w we have $\frac{1}{2} \leq \frac{f_s(w)}{f_{s'}(w)} \leq \frac{3}{2}$ and $r(s, s') \leq \frac{3}{2}$ and $rd(s, s') \leq 2$. $\square$

Lemma 4.7. *If $\mathcal{A} \notin$ EMPTY then $49 < r(s, s'') \leq 51$. If $\mathcal{A} \in$ EMPTY then $r(s, s'') \leq 2$. In either case, $r(s'', s) \leq 2$.*

Proof. We first observe that $\frac{f_{s''}(\vdash w)}{f_s(\vdash w)}$ is always ≤ 2, resulting in the only interesting direction being $\frac{f_s(\vdash w)}{f_{s''}(\vdash w)}$:

$$
\begin{aligned}
\frac{f_{s''}(\vdash w)}{f_s(\vdash w)} &= \frac{\frac{99}{100}f_{s_0}(w) + \frac{1}{100}f_{q_s}(w)}{\frac{1}{2}f_{s_0}(w) + \frac{1}{2}f_{q_s}(w)} \\
&= \frac{\frac{99}{100}f_{s_0}(w)}{\frac{1}{2}f_{s_0}(w) + \frac{1}{2}f_{q_s}(w)} + \frac{\frac{1}{100}f_{q_s}(w)}{\frac{1}{2}f_{s_0}(w) + \frac{1}{2}f_{q_s}(w)} \\
&\leq \frac{\frac{99}{100}f_{s_0}(w)}{\frac{1}{2}f_{s_0}(w)} + \frac{\frac{1}{100}f_{q_s}(w)}{\frac{1}{2}f_{q_s}(w)} \\
&= \frac{2 \cdot 99}{100} + \frac{2}{100} = 2.
\end{aligned}
$$

We observe that for all words $\vdash w$, the quantities r and rd are bounded:

$$
\begin{aligned}
\frac{f_s(\vdash w)}{f_{s''}(\vdash w)} &= \frac{\frac{1}{2}f_{s_0}(w) + \frac{1}{2}f_{q_s}(w)}{\frac{99}{100}f_{s_0}(w) + \frac{1}{100}f_{q_s}(w)} \\
&= \frac{\frac{1}{2}f_{s_0}(w)}{\frac{99}{100}f_{s_0}(w) + \frac{1}{100}f_{q_s}(w)} + \frac{\frac{1}{2}f_{q_s}(w)}{\frac{99}{100}f_{s_0}(w) + \frac{1}{100}f_{q_s}(w)} \\
&\leq \frac{\frac{1}{2}f_{s_0}(w)}{\frac{99}{100}f_{s_0}(w)} + \frac{\frac{1}{2}f_{q_s}(w)}{\frac{1}{100}f_{q_s}(w)} \\
&\leq \frac{100}{2 \cdot 99} + \frac{100}{2} \leq 51.
\end{aligned}
$$

If there is a word w that is accepted by the automaton with probability $> \frac{1}{2}$, then we consider the sequence of words $w_i' = (w \ acc)^i \ rej$:

$$
\frac{f_s(\vdash w_i')}{f_{s''}(\vdash w_i')} = \frac{\frac{1}{2}f_{s_0}(w_i') + \frac{1}{2}f_{q_s}(w_i')}{\frac{99}{100}f_{s_0}(w_i') + \frac{1}{100}f_{q_s}(w_i')} \geq \frac{\frac{1}{2}f_{q_s}(w_i')}{\frac{99}{100}f_{s_0}(w_i') + \frac{1}{100}f_{q_s}(w_i')}.
$$

By the previous proof (Eq. (4.2)) we know $\frac{f_{q_s}(w_i')}{f_{s_0}(w_i')} \xrightarrow[i\to\infty]{} \infty$, thus $\frac{f_{s_0}(w_i')}{f_{q_s}(w_i')} \xrightarrow[i\to\infty]{} 0$. Consider

$$
\frac{\frac{99}{100}f_{s_0}(w_i') + \frac{1}{100}f_{q_s}(w_i')}{\frac{1}{2}f_{q_s}(w_i')} = \frac{2}{100} + \frac{2 \cdot 99}{100}\left(\frac{f_{s_0}(w_i')}{f_{q_s}(w_i')}\right) \xrightarrow[i\to\infty]{} \frac{2}{100}
$$

then

$$
\frac{\frac{1}{2}f_{q_s}(w_i')}{\frac{99}{100}f_{s_0}(w_i') + \frac{1}{100}f_{q_s}(w_i')} \xrightarrow[i\to\infty]{} \frac{100}{2} = 50.
$$

So for all ε there exists an i such that $\frac{f_s(\vdash w_i')}{f_{s''}(\vdash w_i')} \geq 50 - \varepsilon$. In particular for example $r(s, s'') \geq 49$.

If there is no such word then $\mathbb{P}(w) \leq \frac{1}{2}$ for all $w \in \Sigma^*$. We show that the total variation distance will be small. All words starting from state s and s' start with $\vdash$ and are terminated by rej, so in general all words take the form $w = \vdash(w_1 \ acc)\ldots(w_n \ acc)(w_{n+1} \ rej)$. Let us consider the probability of such words from s, s'':

$$\frac{f_s(w)}{f_{s''}(w)} = \frac{\frac{1}{2}f_{s_0}(w') + \frac{1}{2}f_{q_s}(w')}{\frac{99}{100}f_{s_0}(w') + \frac{1}{100}f_{q_s}(w')} \leq \frac{\frac{1}{2}f_{s_0}(w') + \frac{1}{2}f_{q_s}(w')}{\frac{99}{100}f_{s_0}(w')}$$

$$\leq \frac{100}{99} \cdot \left(\frac{1}{2} + \frac{1}{2}\frac{f_{q_s}(w')}{f_{s_0}(w')}\right)$$

$$\leq \frac{100}{99} \cdot \frac{3}{2} \qquad\qquad \text{(by Eq. (4.4))}$$

$$\leq 2.$$

This creates a significant gap between the case where there is a word with probability greater than one half and not; in particular if there exists w with $\mathbb{P}(w) > \frac{1}{2}$ then $49 < r(s, s'') \leq 51$ and $49 < rd(s, s'') \leq 51$, but if no such word exists then $r(s, s'') \leq 2$ and $rd(s, s'') \leq 2$. $\qquad\square$

Remark 4.8. The classic *non-strict* threshold problem for the total variation distance (i.e. whether $tv(s, s') \leq \theta$) is known to be undecidable [Kie18], like our distances. However, it is not known if its strict variant (i.e. whether $tv(s, s') < \theta$) is also undecidable. In contrast, in our case, both variants are undecidable. Further note that (additive) approximation of tv is possible [Kie18, CK14], but this is not the case for our distances r and rd.

5. The relation to the Value-1 Problem

In the previous section we have shown the undecidability of the big-O problem using the undecidability of the emptiness problem for probabilistic automata. Another proof of undecidability can be obtained using the Value-1 problem, shown to be undecidable in [GO10]: indeed, we will show in this section that the big-O problem and the Value-1 problem are interreducible.

The Value-1 problem asks whether there exists a sequence of words with probability tending to 1:

Definition 5.1 (Value-1 problem).
  INPUT   Probabilistic automaton $\mathcal{A}$
  OUTPUT   For all $\delta > 0$, does there exist a word w such that $\mathbb{P}_{\mathcal{A}}(w) > 1 - \delta$?

We will exhibit a close, but not complete, connection between the Value-1 problem and big-O problem by reducing in both directions between the two, see Section 5.1. In this section we are concerned only with matters of decidability, and thus do not consider whether the reductions are efficient.

Our main decidability results (in Sections 7, 8, and 9) are independent of the findings of this section, so it can be skipped during the first reading. However, Section 5.2 below justifies these further developments, showing why the connection exhibited here does not entail decidability results for the big-O problem by a transfer from decidable cases of Value-1.

5.1. **Interreduction between Value-1 and the big-O problem.**

Proposition 5.2. *The* Value-1 *problem reduces to the big-O problem.*

Proof. Assume we are given a probabilistic automaton $\mathcal{A} = \langle Q, \Sigma, M, F_{\mathcal{A}} \rangle$ and a dedicated starting state $q_0 \in Q$, which accepts words w with probability $\mathbb{P}_{\mathcal{A}}(w)$. First construct $\mathcal{A}'$ in which words w are accepted with probability $\mathbb{P}_{\mathcal{A}'}(w) = 1 - \mathbb{P}_{\mathcal{A}}(w)$, by inverting accepting states, that is, $\mathcal{A}' = \langle Q, \Sigma, M, F_{\mathcal{A}'} \rangle$ with $F_{\mathcal{A}'} = Q \setminus F_{\mathcal{A}}$.

Let us first consider the idea of the proof. Consider the probabilistic automaton $\mathcal{B}$ such that $\mathcal{B}(w) = 1$ for all $w \in \Sigma^*$. Then $\mathcal{B}$ is not big-O of $\mathcal{A}'$ if and only if there is a sequence of words for which $\mathcal{A}'(w_i) \xrightarrow{i \to \infty} 0$. Since we wish to show the undecidability even for labelled Markov chains, we adjust the weight at each step by the size of the alphabet in order to encode a probabilistic automaton inside the labelled Markov chain. The remainder of the proof formally builds this construction.

Construct a Markov chain $\mathcal{M}_{\mathcal{A}'} = \langle Q', \Sigma', M', F' \rangle$, where $Q' = Q \cup \{s, s', s_0, rej, acc\}$, $\Sigma' = \Sigma \cup \{\$\}$ (assuming $\$ \notin \Sigma$) and $F' = \{acc\}$. The probabilistic automaton $\mathcal{A}'$ will be simulated by $\mathcal{M}_{\mathcal{A}'}$. The map $M'(a)(q, q') = p$ is described using the notation $q \xrightarrow{p}{a} q'$ as follows:

For all $q \in Q$:

$$\forall q' \in Q : q \xrightarrow{(1/|\Sigma'|)M(a)(q,q')}{a} q' \text{ for all } a \in \Sigma$$

$$\text{if } q \in F_{\mathcal{A}'} : q \xrightarrow{(1/|\Sigma'|)}{\$} acc \quad \text{and} \quad \text{if } q \notin F_{\mathcal{A}'} : q \xrightarrow{(1/|\Sigma'|)}{\$} rej$$

$$s' \xrightarrow{\frac{1}{\$}} q_0 \qquad s \xrightarrow{\frac{1}{\$}} s_0 \qquad s_0 \xrightarrow{(1/|\Sigma'|)}{\$} acc \qquad \text{and} \qquad s_0 \xrightarrow{(1/|\Sigma'|)}{a} s_0 \text{ for all } a \in \Sigma.$$

Note the only words with positive probability are words of the form $\$\Sigma^*\$ \subseteq \Sigma'^*$. Then given a word $w \in \Sigma^*$, $f_s(\$w\$) = \left(\frac{1}{|\Sigma|+1}\right)^{|w\$|}$ and $f_{s'}(\$w\$) = \left(\frac{1}{|\Sigma|+1}\right)^{|w\$|}(1 - \mathbb{P}_{\mathcal{A}}(w))$.

Then if there is a sequence of words $(w_i)_i$ for which $\mathbb{P}_{\mathcal{A}}(w_i)$ tends to 1 then $\frac{f_s(\$w_i\$)}{f_{s'}(\$w_i\$)}$ is unbounded. However, if there exists some $\gamma > 0$ so that for all $w \in \Sigma^*$ we have $\mathbb{P}_{\mathcal{A}}(w) \leq (1-\gamma)$ then $(1 - \mathbb{P}_{\mathcal{A}}(w)) \geq \gamma$, and so $\frac{f_s(\$w\$)}{f_{s'}(\$w\$)} \leq \frac{1}{\gamma}$. $\qquad \square$

Proposition 5.3. *The big-O problem reduces to the* Value-1 *problem.*

Proof. We take an instance of the big-O problem, assumed to be a labelled Markov chain with states s, s' and construct probabilistic automaton $\mathcal{A}'$ as an instance of the Value-1 problem. This new probabilistic automaton will have the property that s is not big-O of s' if and only if $\mathcal{A}'$ is a positive instance of the Value-1 problem. We start with the construction, followed by both directions of the implication.

Construction 5.4. Given a labelled Markov chain $\mathcal{M} = \langle Q, \Sigma, M, F \rangle$ and $s, s' \in Q$, construct a probabilistic automaton $\mathcal{A} = \langle Q', \Sigma', M', F' \rangle$. Each state of Q will be duplicated, once for s and once for s'; $Q_s = \{q_s \mid q \in Q\}$, $Q_{s'} = \{q_{s'} \mid q \in Q\}$. Let $Q' = Q_s \cup Q_{s'} \cup \{q_0, acc, rej, sink_1, sink_2\}$, $\Sigma' = \Sigma \cup \{\$\}$ and $F' = \{acc\}$. The reduction can be seen in Fig. 3.

Each transition of $\mathcal{M}$ will be simulated in each of the copies according to the probability in $\mathcal{M}$. For every $q, q' \in Q$ and $a \in \Sigma$, let $M'(a)(q_s, q'_s) = M(a)(q, q')$ and $M'(a)(q_{s'}, q'_{s'}) = M(a)(q, q')$. A probabilistic automaton should be stochastic for every $a \in \Sigma$, so there is
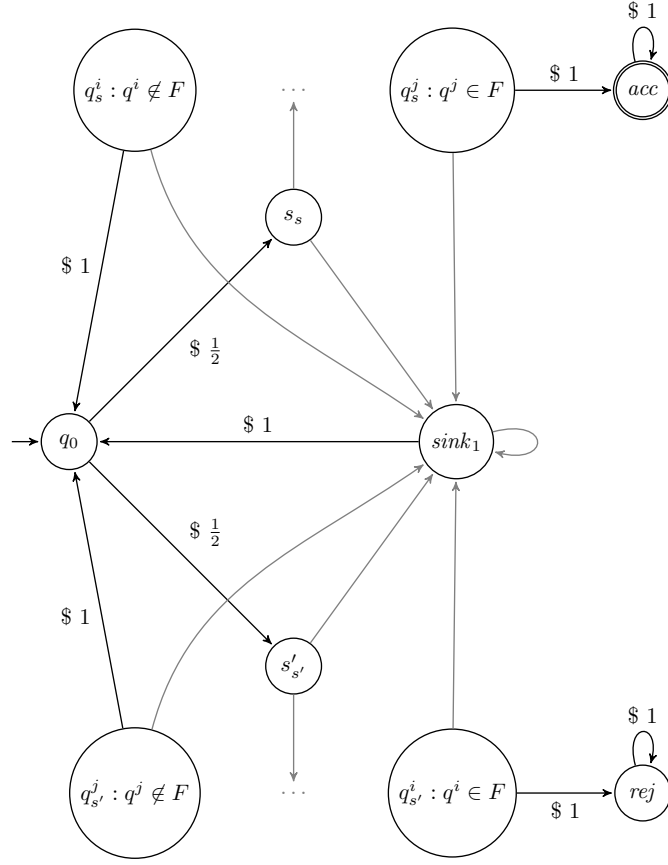
Figure 3. Reduction to Value-1. Only the effect of transitions on the $\$$ symbol are shown in black, with the possibility to transition to the recoverable sink state $sink_1$ depicted in grey (on symbols in Σ). All remaining transitions and $sink_2$ are omitted.

unused probability for each character, which will divert to a (recoverable) sink. For every $q \in Q$ and $a \in \Sigma$, let

$$M'(a)(q_s, sink_1) = 1 - \sum_{q' \in Q} M(a)(q, q')$$

and

$$M'(a)(q_{s'}, sink_1) = 1 - \sum_{q' \in Q} M(a)(q, q').$$

There is an additional control character $\$$: from q_0 the machine will pick either of the two machines with equal probability: $M(\$)(q_0, s_s) = M(\$)(q_0, s'_{s'}) = \frac{1}{2}$. If in the accepting, rejecting, or second (unrecoverable) sink state the system will stay there forever: $M'(a)(acc, acc) = M'(a)(rej, rej) = M'(a)(sink_2, sink_2) = 1$ for all $a \in \Sigma'$ (including $\$$).

The behaviour on $\$$ will differ in the two copies of $\mathcal{M}$. If in an accepting state reached from s, the system will accept. If in an accepting state reached from s' the system will reject. Otherwise the system will restart back to q_0.

Formally,

$$M'(\$)(q_s, acc) = 1 \text{ when } q_s \in F \text{ and } M'(\$)(q_s, q_0) = 1 \text{ when } q_s \notin F$$

and

$$M'(\$)(q_{s'}, rej) = 1 \text{ when } q_{s'} \in F \text{ and } M'(\$)(q_{s'}, q_0) = 1 \text{ when } q_{s'} \notin F.$$

When in the recoverable sink state, the system restarts on $\$$, $M'(\$)(sink_1, q_0) = 1$, and stays there for all $a \in \Sigma$, $M'(a)(sink_1, sink_1) = 1$.

It is intended that all $a \in \Sigma$ are unreadable from q_0: the system moves to the unrecoverable sink state $sink_2$, $M'(a)(q_0, sink_2) = 1$.

The idea is that if $f_s(w) \gg f_{s'}(w)$ then, by repeatedly reading the word w, all of the probability mass will eventually move to acc; otherwise a sufficiently large amount of mass will be lost to rej.

Denote by $\mathbb{P}_{\mathcal{A}}(w)$ the probability of a word w in the probabilistic automaton $\mathcal{A}$, from state q_0. Here we use the notation f to refer to the probability in the original labelled Markov chain $\mathcal{M}$. Further, the notation $\mathbb{P}[q \xrightarrow{w} q']$ stands for $\left(M'(w_1) \times \cdots \times M'(w_{|w|})\right)_{q,q'}$, i.e., the probability of transitioning from state q to q' while reading w in $\mathcal{A}$.

Direction 1 (Not big-O implies VALUE-1). The proof shows that for every $\delta > 0$ there exists $C > 0$ such that, for all $w \in \Sigma^*$, the inequality $f_s(w) > Cf_{s'}(w)$ implies $\mathbb{P}_{\mathcal{A}}((\$w\$)^i) > 1 - \delta$ for some appropriately chosen $i \in \mathbb{N}$.

Hence given δ, choose C such that $(1 - \frac{\delta}{2})\frac{C}{C+1} > 1 - \delta$. Then by the not big-O property, choose a word w such that $f_s(w) = C'f_{s'}(w)$, with $C' > C$. Then $(1-\frac{\delta}{2})\frac{C'}{C'+1} > (1-\frac{\delta}{2})\frac{C}{C+1} > 1 - \delta$.

The input word $\$w\$$ induces a (unary) Markov chain, represented by the matrix A, representing states q_0, acc and rej in the three positions, respectively:

$$A = \begin{bmatrix} 0.5(1 - f_s(w)) + 0.5(1 - f_{s'}(w)) & 0.5f_s(w) & 0.5f_{s'}(w) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then, for the word $(\$w\$)^i$, for $i \in \mathbb{N}$, starting from state q_0, observe:

$$A^i \xrightarrow[i \to \infty]{} \begin{bmatrix} 0 & C'x & x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ with } C'x + x = 1.$$

For each i, we have $A^i(q_0, acc) + A^i(q_0, rej) + A^i(q_0, q_0) = 1$. Choose i such that $A^i(q_0, q_0) \leq \frac{\delta}{2}$. Then $A^i(q_0, acc) + A^i(q_0, rej) \geq 1 - \frac{\delta}{2}$, using the fact that $A^i(q_0, acc) = C'A^i(q_0, rej)$, obtaining $A^i(q_0, acc) + \frac{A^i(q_0,acc)}{C'} \geq 1 - \frac{\delta}{2}$. Hence $\mathbb{P}_{\mathcal{A}}((\$w\$)^i) = A^i(q_0, acc) \geq (1 - \frac{\delta}{2})\frac{C'}{C'+1} > 1 - \delta$, as required.

Direction 2 (big-O implies Not VALUE-1). We know that there exists $C > 0$ such that, for all $w \in \Sigma^*$, $f_s(w) \leq Cf_{s'}(w)$, and should show that there exists $\delta > 0$ such that every $w \in (\Sigma \cup \{\$\})^*$ has $\mathbb{P}_{\mathcal{A}}(w) \leq 1 - \delta$.

To move probability from q_0 to acc it is necessary to use words of the form $(\$\Sigma^*\$)^*$ where Σ is the alphabet of $\mathcal{M}$. Hence any word w can be decomposed into $\$w_m\$\$w_{m-1}\$...\$w_1\$$.

Let us define $x_i = \mathbb{P}[q_0 \xrightarrow{\$w_i\$...\$w_1\$} acc]$ and $y_i = \mathbb{P}[q_0 \xrightarrow{\$w_i\$...\$w_1\$} rej]$, and observe $x_m = \mathbb{P}_{\mathcal{A}}(w)$. We compute x_i, y_i inductively and show $x_i \leq Cy_i$ for all $1 \leq i \leq m$.

Consider reading w_1, the probability is such that

$$x_1 = \mathbb{P}[q_0 \xrightarrow{\$w_1\$} acc] = \frac{1}{2}f_s(w_1)$$

$$y_1 = \mathbb{P}[q_0 \xrightarrow{\$w_1\$} rej] = \frac{1}{2}f_{s'}(w_1)$$

$$\mathbb{P}[q_0 \xrightarrow{\$w_1\$} q_0] = 1 - x_1 - y_1.$$

Since there exists $C > 0$ such that, for all w_i, $f_s(w_i) \leq Cf_{s'}(w_i)$, we have $x_1 \leq Cy_1$. Repeating this argument inductively, we show that $x_i \leq Cy_i$ for all i. Indeed:

$$x_i = \mathbb{P}[q_0 \xrightarrow{\$w_i\$...\$w_1\$} acc]$$

$$= \mathbb{P}[q_0 \xrightarrow{\$w_i\$} q_0] \cdot \mathbb{P}[q_0 \xrightarrow{\$w_{i-1}\$...\$w_1\$} acc] + \mathbb{P}[q_0 \xrightarrow{\$w_i\$} acc]$$

$$= (1 - \frac{1}{2}f_s(w_i) - \frac{1}{2}f_{s'}(w_i)) \cdot x_{i-1} + \frac{1}{2}f_s(w_i)$$

$$y_i = \mathbb{P}[q_0 \xrightarrow{\$w_i\$...\$w_1\$} rej]$$

$$= \mathbb{P}[q_0 \xrightarrow{\$w_i\$} q_0] \cdot \mathbb{P}[q_0 \xrightarrow{\$w_{i-1}\$...\$w_1\$} rej] + \mathbb{P}[q_0 \xrightarrow{\$w_i\$} rej]$$

$$= (1 - \frac{1}{2}f_s(w_i) - \frac{1}{2}f_{s'}(w_i)) \cdot y_{i-1} + \frac{1}{2}f_{s'}(w_i).$$

Hence, assuming $f_s(w_i) \leq Cf_{s'}(w_i)$ and $x_{i-1} \leq Cy_{i-1}$ we have:

$$x_i = (1 - \frac{1}{2}f_s(w_i) - \frac{1}{2}f_{s'}(w_i)) \cdot x_{i-1} + \frac{1}{2}f_s(w_i)$$

$$\leq (1 - \frac{1}{2}f_s(w_i) - \frac{1}{2}f_{s'}(w_i)) \cdot Cy_{i-1} + C \cdot \frac{1}{2}f_{s'}(w_i)$$

$$= C \cdot [(1 - \frac{1}{2}f_s(w_i) - \frac{1}{2}f_{s'}(w_i))y_{i-1} + \frac{1}{2}f_{s'}(w_i)]$$

$$\leq Cy_i.$$

We have $x_m \leq Cy_m \leq C(1 - x_m) = C - Cx_m$, so $\mathbb{P}_{\mathcal{A}}(w) = x_m \leq \frac{C}{C+1} < 1$, so the probability of reaching acc is bounded away from 1 for every word. $\qquad\square$

5.2. What do decidable cases of the Value-1 problem tell us about the big-O problem?

The Value-1 problem is undecidable in general, however it is decidable in the unary case in **coNP** [CKV14] and for *leaktight automata* [FGO12]. Note, however, that the construction of Proposition 5.3 combined with these decidability results *does not* entail decidability results for the big-O problem. Firstly note that the construction adds an additional character, and so a unary instance of the big-O problem always has at least two characters when translated to the Value-1 problem. Further, as we show in the remainder of this section, the construction does not result in a leaktight automaton. The following argument, does not, of course, preclude the existence of an alternative construction which could, hypothetically, maintain these properties.

Let us recall the definition of leaktight automata [FGO12].

Definition 5.5. A finite word u is *idempotent* if reading once or twice the word u does not change qualitatively the transition probabilities. That is $\mathbb{P}_\mathcal{A}[q \xrightarrow{u} q'] > 0 \iff \mathbb{P}_\mathcal{A}[q \xrightarrow{uu} q'] > 0$.

Let u_n be a sequence of idempotent words. Assume that the sequence of matrices $\mathbb{P}_\mathcal{A}(u_n)$ converges to a limit M, that this limit is idempotent and denote $\mathcal{M}$ the associated Markov chain. The sequence u_n is a *leak* if there exist $r, q \in Q$ such that the following three conditions hold:

(1) r and q are recurrent in M,
(2) $\lim_{n\to\infty} \mathbb{P}_\mathcal{A}[r \xrightarrow{u_n} q] = 0$,
(3) for all n, $\mathbb{P}_\mathcal{A}[r \xrightarrow{u_n} q] > 0$.

An automaton is *leaktight* if there is no leak.

If there were no leak in the probabilistic automata obtained from the reduction of Proposition 5.3, then the decidability of the big-O problem for labelled Markov chains would follow. However, this is not the case, i.e., Proposition 5.3 does not solve any cases of the big-O problem by reducing to known decidable fragments of VALUE-1.

Proposition 5.6. *The resulting automaton from the reduction of the big-O problem to the* VALUE-1 *problem has a leak.*

Proof. Consider some infinite sequence of words w_i growing in length, such that $f_s(w_i) > 0$ for every i. Let $u_i = \$w_i\$$.

Observe that this word is idempotent. For each starting state, consider the possible states with non-zero probability and from each of these the set of reachable states. Observe that in all cases the set reachable after one application is equal to the set reachable after two:

- $acc \xrightarrow{\$w_i\$} acc \xrightarrow{\$w_i\$} acc$

- $rej \xrightarrow{\$w_i\$} rej \xrightarrow{\$w_i\$} rej$

- $q_0 \xrightarrow{\$w_i\$} q_0, acc, rej \xrightarrow{\$w_i\$} q_0, acc, rej$

- For q accepting in Q_s: $q \xrightarrow{\$w_i\$} acc \xrightarrow{\$w_i\$} acc$

- For q rejecting in Q_s: $q \xrightarrow{\$w_i\$} sink_2 \xrightarrow{\$w_i\$} sink_2$

- For q accepting in $Q_{s'}$: $q \xrightarrow{\$w_i\$} rej \xrightarrow{\$w_i\$} rej$

- For q rejecting in $Q_{s'}$: $q \xrightarrow{\$w_i\$} sink_2 \xrightarrow{\$w_i\$} sink_2$

- $sink_1 \xrightarrow{\$w_i\$} sink_2 \xrightarrow{\$w_i\$} sink_2$

- $sink_2 \xrightarrow{\$w_i\$} sink_2 \xrightarrow{\$w_i\$} sink_2$.

Assume that the original labelled Markov chain has a sink, that is, the decision to terminate the word must be made by probability. Then for all $\lambda > 0$ there exists n such that $f_s(\Sigma^{>n}) < \lambda$ and $f_{s'}(\Sigma^{>n}) < \lambda$ [Kie18, Lemma 12].

Suppose the sequence of matrices $\mathbb{P}_\mathcal{A}(u_n)$ converges to a limit M (this is with no loss of generality, because each matrix in the sequence belongs to a compact set, so a convergent subsequence must exist) and let $r = q_0$ and $q = acc$.
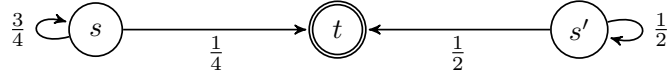
Figure 4. Unbounded ratio but language equivalent.

By the observation above, for longer and longer words the probability of reaching *acc* from q_0 is diminishing. Thus $\lim \mathbb{P}_{\mathcal{A}}[r \xrightarrow{u_n} q] = 0$, and in M we have r and q in different SCCs. The state *acc* is recurrent as it is deterministically looping on every character. Since the probability of reaching a final state is diminishing for longer and longer words, whenever $ is read the state returns to r, hence all words return to r with probability 1 in the limit. By the choice of words in the sequence, for every word $f_s(w_n) > 0$, we have $\mathbb{P}_{\mathcal{A}}[r \xrightarrow{u_n} q] > 0$ for all n.

Hence a leak has been defined, even in the case where $\mathcal{M}$ is unary. $\square$

6. The Language Containment condition

Towards decidability results, we identify a simple necessary (but insufficient) condition for s being big-O of s'.

Definition 6.1 (LC condition). A weighted automaton $\mathcal{W} = \langle Q, \Sigma, M, F \rangle$ and $s, s' \in Q$ satisfy the *language containment condition* (LC) if for all words w with $f_s(w) > 0$ we also have $f_{s'}(w) > 0$. Equivalently, $\mathcal{L}_s(\mathcal{W}) \subseteq \mathcal{L}_{s'}(\mathcal{W})$.

The condition can be verified by constructing NFA $\mathcal{N}_s(\mathcal{W}), \mathcal{N}_{s'}(\mathcal{W})$ that accept $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ respectively and verifying $\mathcal{L}(\mathcal{N}_s(\mathcal{W})) \subseteq \mathcal{L}(\mathcal{N}_{s'}(\mathcal{W}))$.

Remark 6.2. Recall that NFA language containment is **NL**-complete if the automata are in fact deterministic, in **P** if they are unambiguous [Col15, Theorem 3], **coNP**-complete if they are unary [SM73] and **PSPACE**-complete in general [MS72]. In all cases this complexity level will match, or be lower than that for our respective algorithm for the big-O problem.

We observe that, if s is big-O of s', the LC condition must hold and so the LC condition is the first step in each of our verification routines. Example 6.3 shows that the condition alone is not sufficient to solve the big-O problem, because two states can admit the same set of words with non-zero weight, yet the weight ratios become unbounded.

Example 6.3. Consider the unary automaton $\mathcal{W}$ in Fig. 4. We have $\mathcal{L}_s(\mathcal{W}) = \mathcal{L}_{s'}(\mathcal{W}) = \{a^n \mid n \geq 1\}$, but $\frac{f_s(a^n)}{f_{s'}(a^n)} = \frac{(0.75)^{n-1} \cdot 0.25}{(0.5)^{n-1} \cdot 0.5} = 0.5 \cdot 1.5^{n-1} \xrightarrow[n \to \infty]{} \infty$.

6.1. **The Eventually Big-O Problem.** Recall from Section 3.3 that the eventually big-O problem asks whether there exists $C > 0, k > 0$ such that forall $w \in \Sigma^{\geq k}$ we have $f_s(w) \leq C \cdot f_{s'}(w)$. We now justify our focus on the Big-O problem by showing the close relationship between the two problems.

The big-O problem reduces to its eventual variant by checking both the LC condition and the eventually big-O condition. Thus our undecidability (and hardness) results transfer to the eventually big-O problem. The eventually big-O problem can be solved via the big-O

problem by "fixing" the LC condition through the addition of a branch from s' that accepts all appropriate words with very low probability.

Let $\mathcal{W} = \langle Q, \Sigma, M, \{t\} \rangle$ be a weighted automaton, $s, s' \in Q$, and $s \neq s'$. Below, whenever we write f_s (resp. $f_{s'}$), this will refer to word weights from s (resp. s') in $\mathcal{W}$. We assume without loss of generality that the state s' has no incoming transitions (if this is not the case, a fresh copy of s' can be made).

Choose δ to be a real number such that $0 < \delta < 1$ and δ is smaller than any positive weight in $\mathcal{W}$. Construct $\mathcal{W}'$ by adding the following transitions for all $x \in \Sigma$:

$$s' \xrightarrow[x]{\delta} t \qquad s' \xrightarrow[x]{\delta} \bullet \qquad \bullet \xrightarrow[x]{\delta} \bullet \qquad \bullet \xrightarrow[x]{\delta} t,$$

where $\bullet$ is a new state. Consequently, for any $w \in \Sigma^+$, we get:

(1) the weight of w in $\mathcal{W}'$ from s' is $f_{s'}(w) + \delta^{|w|}$,
(2) if $f_{s'}(w) > 0$ then $f_{s'}(w) > \delta^{|w|}$.

Proposition 6.4. *s is eventually big-O of s' in $\mathcal{W}$ if and only if $\mathcal{L}_s(\mathcal{W}) \setminus \mathcal{L}_{s'}(\mathcal{W})$ is finite and s is big-O of s' in $\mathcal{W}'$.*

Proof.

Direction 1 ($\Rightarrow$). Suppose s is eventually big-O of s' in $\mathcal{W}$, i.e. there exist C, k such that, for all $w \in \Sigma^{\geq k}$, $f_s(w) \leq C f_{s'}(w)$. Note that, for $w \in \Sigma^{\geq k}$, this implies that, whenever $f_s(w) > 0$, we must also have $f_{s'}(w) > 0$. Consequently, $\mathcal{L}_s(\mathcal{W}) \setminus \mathcal{L}_{s'}(\mathcal{W}) \subseteq \Sigma^{<k}$, i.e. $\mathcal{L}_s(\mathcal{W}) \setminus \mathcal{L}_{s'}(\mathcal{W})$ must be finite.

Let $w \in \Sigma^*$, $m = \max_{w \in \Sigma^{<k}} f_s(w)$ and $C' = \frac{m}{\delta^k}$.

- If $w \in \Sigma^{\geq k}$, we have $f_s(w) \leq C f_{s'}(w) \leq C(f_{s'}(w) + \delta^{|w|})$.
- If $w \in \Sigma^{<k}$, then $f_s(w) \leq m = \frac{m}{\delta^k} \delta^k = C' \delta^k \leq C' \delta^{|w|} \leq C'(f_{s'}(w) + \delta^{|w|})$. Note that $\delta^k \leq \delta^{|w|}$ follows from $w \in \Sigma^{<k}$ and $0 < \delta < 1$.

Taking $\max(C, C')$ as the relevant constant, we can conclude that s is big-O of s' in $\mathcal{W}'$.

Direction 2 ($\Leftarrow$). Suppose $\mathcal{L}_s(\mathcal{W}) \setminus \mathcal{L}_{s'}(\mathcal{W})$ is finite and s is big-O of s' in $\mathcal{W}'$. Because $\mathcal{L}_s(\mathcal{W}) \setminus \mathcal{L}_{s'}(\mathcal{W})$ is finite, there exists k such that, for all $w \in \Sigma^{\geq k}$, $f_s(w) > 0$ implies $f_{s'}(w) > 0$. Because s is big-O of s' in $\mathcal{W}'$, there exists C such that $f_s(w) \leq C(f_{s'}(w) + \delta^{|w|})$ for any $w \in \Sigma^*$.

Let $w \in \Sigma^{\geq k}$. From s being big-O of s', we get $f_s(w) \leq C(f_{s'}(w) + \delta^{|w|})$.

- If $f_s(w) > 0$ then $f_{s'}(w) > 0$. By construction of $\mathcal{W}'$, we get $f_{s'}(w) > \delta^{|w|}$, so

$$f_s(w) \leq C(f_{s'}(w) + \delta^{|w|}) < C(f_{s'}(w) + f_{s'}(w)) = 2C f_{s'}(w).$$

- If $f_s(w) = 0$ then we also have $f_s(w) = 0 \leq 2C f_{s'}(w)$.

Consequently, for any $w \in \Sigma^{\geq k}$, $f_s(w) \leq 2C f_{s'}(w)$, i.e. s is eventually big-O of s' in $\mathcal{W}$. $\square$

The above argument relied on completing the automaton so that any word is accepted with some weight. To transfer our decidability results for bounded languages, it will be necessary to complete the automaton with respect to a bound, i.e. the extra weights are added only for words from $a_1^+ \cdots a_m^+$, $a_1^* \cdots a_m^*$, $w_1^* \cdots w_k^*$ respectively. This can be done easily by introducing the extra transitions according to DFA for the bounding language.

7. The big-O problem for unary weighted automata is **coNP**-complete

In this section we show **coNP**-completeness in the unary case.

Theorem 7.1. *The big-O problem for unary weighted automata is* **coNP**-*complete. It is* **coNP**-*hard even for unary labelled Markov chains.*

For the upper bound, our analysis will refine the analysis of the growth of powers of non-negative matrices of Friedland and Schneider [FS80, Sch86] which gives the asymptotic order of growth of $A_{s,t}^n + A_{s,t}^{n+1} + \cdots + A_{s,t}^{n+q} \approx \rho^n n^k$ for some ρ, k and q, which smooths over the periodic behaviour (see Theorem 7.5). Our results require a non-smoothed analysis, valid for each n. This is not provided in [FS80, Sch86], where the smoothing forces the existence of a single limit—which we don't require. Our big-Θ lemma (Lemma 7.9) will accurately characterise the asymptotic behaviour of $A_{s,t}^n$ by exhibiting the correct value of ρ and k for every word.

We show that these asymptotic behaviours can be captured using suitably defined finite automata. We then reduce the big-O problem to language containment problems on these automata. We start by introducing several definitions, and the result of Friedland and Schneider in Section 7.1, before implementing the proof of the upper bound in Section 7.2. We show **coNP**-hardness in Section 7.3 to complete the proof of Theorem 7.1.

7.1. **Preliminaries.** Let $\mathcal{W}$ be a unary non-negative weighted automaton with states Q, transition matrix A and a unique final state t. When we refer to a *path* in $\mathcal{W}$, we mean a path in the NFA of $\mathcal{W}$, i.e. paths only use transitions with non-zero weights and states on a path may repeat.

Definition 7.2.
- A state q can *reach* q' if there is a path from q to q'. In particular, any state q can always reach itself.
- A *strongly connected component* (SCC) $\varphi \subseteq Q$ is a maximal set of states such that for each $q, q' \in \varphi$, q can reach q'. We denote by $SCC(q)$ the SCC of state q and by A^φ the $|\varphi| \times |\varphi|$ transition matrix of φ. Note every state is in a SCC, even if it is a singleton.
- The DAG of $\mathcal{W}$ is the directed acyclic graph of strongly connected components. Components φ, φ' are connected by an edge if there exist $q \in \varphi$ and $q' \in \varphi'$ with $A(q, q') > 0$.
- The spectral radius of an $m \times m$ matrix A is the largest absolute value of its eigenvalues. Recall the eigenvalues of A are $\{\lambda \in \mathbb{C} \mid \text{ exists vector } \vec{x} \in \mathbb{C}^m, \vec{x} \neq 0 \text{ with } A\vec{x} = \lambda\vec{x}\}$. The spectral radius of φ, denoted by ρ_φ, is the spectral radius of A^φ. By $\rho(q)$ we denote the spectral radius of the SCC in which q is a member.
- We denote by T^φ the period of the SCC φ: the greatest common divisor of return times for some state $s \in \varphi$, i.e. $\gcd\{t \in \mathbb{N} \mid A^t(s, s) > 0\}$. It is known that any choice of state in the SCC gives the same value (see e.g., [Ser13, Theorem 1.20]). If $A^\varphi = [0]$ then $T^\varphi = 0$.
- Let $\mathscr{P}(s, s')$ be the set of paths from the SCC of s to the SCC of s' in the DAG of $\mathcal{W}$. Thus a path $\pi \in \mathscr{P}(s, s')$ is a sequence of SCCs $\varphi_1, \ldots, \varphi_m$.
- $T(s, s')$, called the local period between s and s', is defined by $T(s, s') = \operatorname*{lcm}_{\pi \in \mathscr{P}(s,s')} \gcd_{\varphi \in \pi} T^\varphi$.
- The spectral radius between states s and s', written $\rho(s, s')$, is the largest spectral radius of any SCC seen on a path from s to s': $\rho(s, s') = \max_{\pi \in \mathscr{P}(s,s')} \rho(\pi)$, where $\rho(\pi) = \max_{\varphi \in \pi} \rho_\varphi$ for $\pi \in \mathscr{P}(s, s')$.

- The following function captures the number of SCCs which attain the largest spectral radius on the path that has the most SCCs of maximal spectral radius. Let $k(s, s') = \max_{\pi \in \mathscr{P}(s,s')} k(\pi) - 1$, where, for $\pi \in \mathscr{P}(s, s')$, $k(\pi) = |\{\varphi \in \pi \mid \rho_\varphi = \rho(s, s')\}|$.

Lemma 7.3. *Given A^φ, a representation of the value ρ_φ can be found in polynomial time. This representation will admit polynomial time testing of $\rho_\varphi > \rho_{\varphi'}$ and $\rho_\varphi = \rho_{\varphi'}$ and can be embedded into the first order theory of the reals.*

Proof. An algebraic number z can be represented as a tuple $(p_z, a, b, r) \in \mathbb{Q}[x] \times \mathbb{Q}^3$. Here p_z is a polynomial over x and a, b, r form an approximation such that z is the only root of $p_z(x)$ with $|z - (a + bi)| \leq r$.

Given a polynomial, one can find the representation of each of its roots in polynomial time using the root separation bound due to Mignotte [Mig82]. Operations such as addition and multiplication of two algebraic numbers, finding $|x|$, testing if $x > 0$ can be done in polynomial time in the size of the representation (p, a, b, r), yielding the same representation [BPR05, Coh13, Pan96]; see, e.g., [OW14, Section 3] for a summary.

Any coefficient of the characteristic polynomial of an *integer* matrix can be found in **GapL** [HT03]. **GapL** is the difference of two **#L** calls, each of which can be found in $\mathbf{NC}^2 \subseteq \mathbf{P}$. Here the matrix will be rational; but it can be normalised to an integer matrix by a scalar, the least common multiple of the denominator of each rational. Whilst this number could be exponential, it is representable in polynomial space. The final eigenvalues can be renormalised by this constant.

The characteristic polynomial of an $n \times n$ matrix has degree at most n, since each coefficient can be found in polynomial time, the whole characteristic polynomial can be found in this time. Thus by enumerating its roots (at most n), taking the modulus of each, and sorting them ($a > b \iff a + (-1) \times b > 0$) we can find the spectral radius in this form (p_z, a, b, r).

Note that the spectral radius is a real number, so that given the spectral radius in the form (p_z, a, b, r) we actually have $b = 0$. Then the number can be encoded exactly in the first order theory of the reals using $\exists x : p_z(x) = 0 \wedge x - a \leq r \wedge a - x \leq r$. $\square$

The asymptotic behaviours of weighted automata will be characterised using (ρ, k)-pairs:

Definition 7.4. A (ρ, k)-pair is an element of $\mathbb{R} \times \mathbb{N}$. The ordering on $\mathbb{R} \times \mathbb{N}$ is lexicographic, i.e. $(\rho_1, k_1) \leq (\rho_2, k_2) \iff \rho_1 < \rho_2 \vee (\rho_1 = \rho_2 \wedge k_1 \leq k_2)$.

Friedland and Schneider [FS80, Sch86] essentially use (ρ, k)-pairs to show the asymptotic behaviour of the powers of non-negative matrices. In particular they find the asymptotic behaviour of the sum of several $A^n_{s,s'}$, smoothing the periodic behaviour of the matrix.

Theorem 7.5 (Friedland and Schneider [FS80, Sch86]). *Let A be an $m \times m$ non-negative matrix, inducing a unary weighted automaton $\mathcal{W}$ with states $Q = \{1, \ldots, m\}$. Given $s, t \in Q$, let $B^n_{s,t} = A^n_{s,t} + A^{n+1}_{s,t} + \cdots + A^{n+T(s,t)-1}_{s,t}$. Then $\lim_{n \to \infty} \frac{B^n_{s,t}}{\rho(s,t)^n n^{k(s,t)}} = c$, $0 < c < \infty$.*

In the case where the local period is 1 ($T(s,t) = T(s',t) = 1$), Theorem 7.5 can already be used to solve the big-O problem (in particular if the matrix A is aperiodic). In this case $A^n_{s,t} = B^n_{s,t} = \Theta(\rho(s,t)^n n^{k(s,t)})$. Then to establish that s is big-O of s' we check that the language containment condition holds and that $(\rho(s,t), k(s,t)) \leq (\rho(s',t), k(s',t))$. However, this is not sufficient if the local period is not 1.
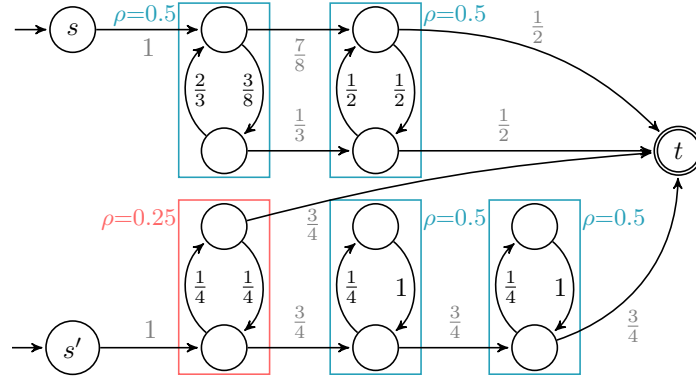
Figure 5. Different rates for different phases.

Example 7.6. Consider the chains shown in Fig. 5 with local period 2. The behaviour for $n \geq 3$ is $A_{s,t}^n = \Theta(0.5^n n)$ and $A_{s',t}^n = \Theta(0.25^n)$ when n is odd and $A_{s',t}^n = \Theta(0.5^n n)$ when n is even. However, Theorem 7.5 tells us $B_{s,t}^n = \Theta(0.5^n n)$ and $B_{s',t}^n = \Theta(0.5^n n)$ suggesting the ratio is bounded, but in fact s is not big-O s' (although s' is big-O of s) because $\frac{A_{s,t}^{2n+1}}{A_{s',t}^{2n+1}} \xrightarrow{n \to \infty} \infty$.

7.2. **Upper bound: The unary big-O problem is in coNP.** Let $\mathcal{W}$ be a unary weighted automaton and suppose we are asked whether s is big-O of s'. We assume w.l.o.g. (a) that there is a unique final state t with no outgoing transitions, and (b) that s, s' do not appear on any cycle (if this is not the case, copies of s, s' and their transitions can be taken).

Next we define a "degree function", which captures the asymptotic behaviour of each word a^n by a (ρ, k)-pair, capturing the exponential and polynomial behaviours respectively.

Definition 7.7. Given a unary weighted automaton $\mathcal{W}$, let $d_{s,t} : \mathbb{N} \to \mathbb{R} \times \mathbb{N}$ be defined by $d_{s,t}(n) = (\rho, k)$, where:

- ρ is the largest spectral radius of any vertex visited on any path of length n from s to t;
- the path from s to t that visits the most SCCs of spectral radius ρ visits $k+1$ such SCCs;
- if there is no length-n path from s to t, then $(\rho, k)=(0,0)$.

The set of *admissible* (ρ, k)-pairs is the image of $d_{s,t}$. Observe that this set is finite and of size at most $|Q|^2$: there can be no more than $|Q|$ values of ρ (if at worst each state were its own SCC) and the value of k is also bounded by the number of SCCs and thus $|Q|$.

To prepare for the proof of our key big-Θ lemma (stated below), we next define the (ρ, k)-annotated version of the weighted automaton $\mathcal{W}$. Namely, in each state we record the relevant value of (ρ, k) corresponding to the current run to the state.

Definition 7.8 (The weighted automaton $\mathcal{W}^\dagger$). Given $\mathcal{W} = \langle Q, \Sigma, A, \{t\} \rangle$ and $s \in Q$, the weighted automaton $\mathcal{W}^\dagger$ has states of the form (q, ρ, k) for all $q \in Q$ and all admissible (ρ, k)-pairs, the same Σ and no final states. For every transition $q \xrightarrow{p} q'$ from $\mathcal{W}$ denoting $A(q, q') = p$, include the following transition in $\mathcal{W}^\dagger$ for each admissible (ρ, k):

- $(q, \rho, k) \xrightarrow{p} (q', \rho, k)$ if $SCC(q){=}SCC(q')$,

- $(q, \rho, k) \xrightarrow{p} (q', \rho, k+1)$ if $SCC(q){\neq}SCC(q')$ and $\rho = \rho(q')$,

- $(q, \rho, k) \xrightarrow{p} (q', \rho, k)$ if $SCC(q){\neq}SCC(q')$ and $\rho > \rho(q')$,

- $(q, \rho, k) \xrightarrow{p} (q', \rho(q'), 0)$ if $SCC(q){\neq}SCC(q')$ and $\rho(q') > \rho$.

Observe that the automaton $\mathcal{W}^\dagger$ is constructable in polynomial time given $\mathcal{W}$. Indeed, the spectral radii of all SCCs can be computed and compared to each other in time polynomial in the size of $\mathcal{W}$ (see Lemma 7.3).

Let $s, t \in Q$ be fixed. We are now ready to state and prove the key technical lemma of this subsection (cf. Theorem 7.5, Friedland and Schneider [FS80, Sch86]), where we assume the functions $\rho(n), k(n)$, defined by $d_{s,t}(n) = (\rho(n), k(n))$.

Lemma 7.9 (The big-Θ lemma). *There exist $c, C > 0$ such that, for every $n > |Q|$,*

$$c \cdot \rho(n)^n n^{k(n)} \leq A_{s,t}^n \leq C \cdot \rho(n)^n n^{k(n)}.$$

Proof.

Direction 1 (lower bound). Let us fix (ρ', k') and show the bound for the elements of $\{n \in \mathbb{N} \mid d_{s,t}(n) = (\rho', k')\}$.

A witnessing path in $\mathcal{W}$ is a length-n path from s to t that visits $k' + 1$ SCCs of spectral radius ρ' and no SCC with a larger spectral radius. Let $\pi = \varphi_1 \ldots \varphi_k \in \mathscr{P}(s, t)$ be the corresponding sequence of SCCs visited by some witnessing path and let s_i, e_i $(1 \leq i \leq k)$ be the entry and exit points (respectively into and out of φ_i) on that path, i.e. $s = s_1$, $SCC(s_i) = SCC(e_i) = \varphi_i$ $(1 \leq i \leq k)$, there is a transition (of positive weight) from e_i to s_{i+1} and $e_k = t$. We write $s_i, \vec{e}_i$ to represent the particular sequence of entry/exit points.

Let us define a new unary weighted automaton $\mathcal{W}^{s_i, \vec{e}_i}$ to be a restriction of $\mathcal{W}$ so that the only transitions between its SCCs are from e_i to s_{i+1}, for $1 \leq i < k$, i.e., the weight is reduced to zero for any violating transition. There are finitely many such $s_i, \vec{e}_i$, so that if $n \in \{n \in \mathbb{N} \mid d_{s,t}(n) = (\rho', k')\}$ then there exists some $\mathcal{W}^{s_i, \vec{e}_i}$ with transition matrix D such that $D_{s,t}^n > 0$.

Let us restrict to a single choice of $s_i, \vec{e}_i$, and fix D be the transition matrix of $\mathcal{W}^{s_i, \vec{e}_i}$. Clearly $A_{s,t}^n \geq D_{s,t}^n$ for all n, since $\mathcal{W}^{s_i, \vec{e}_i}$ is a restriction of $\mathcal{W}$. We now show that $D_{s,t}^n \geq c_{s_i, \vec{e}_i} (\rho')^n n^{k'}$ for some $c_{s_i, \vec{e}_i} > 0$ whenever $D_{s,t}^n > 0$. Hence, this will imply that $A_{s,t}^n \geq c_{s_i, \vec{e}_i} (\rho')^n n^{k'}$.

Note that, in $\mathcal{W}^{s_i, \vec{e}_i}$, $\rho(s, t) = \rho'$ and $k(s, t) = k'$, because all paths from s to t must visit $k' + 1$ SCC's with spectral radius ρ'. Therefore, by Theorem 7.5 there exists $c' > 0$ be such that

$$\lim_{m \to \infty} \frac{D_{s,t}^m + D_{s,t}^{m+1} + \cdots + D_{s,t}^{m+T-1}}{(\rho')^m m^{k'}} = c',$$

where T is the local period from s to t in $\mathcal{W}^{s_i, \vec{e}_i}$. By the definition of limit, for all $\varepsilon > 0$ the ratio is at least $c' - \varepsilon$ if m is big enough. Pick $\varepsilon = c'/2$, then the inequality $D_{s,t}^m + D_{s,t}^{m+1} + \cdots + D_{s,t}^{m+T-1} \geq (c'/2) \cdot (\rho')^m m^{k'}$ holds for all $m \geq N$ where N is a constant dependent only on $s_i, \vec{e}_i$.

Next we shall show that whenever $D_{s,t}^n > 0$, that is the length n word has a positive path from s to t in $\mathcal{W}^{s_i, \vec{e}_i}$, we have $D_{s,t}^{n+1} + \cdots + D_{s,t}^{n+T-1} = 0$. This will imply

$D^n_{s,t} \geq (c'/2)(\rho')^n n^{k'}$ (if $n > N$) and, hence, $A^n_{s,t} \geq (c'/2)(\rho')^n n^{k'}$. Let L be the length of the shortest path from s to t in $\mathcal{W}^{s_i,\vec{e}_i}$. Observe that paths from s to t in $\mathcal{W}^{s_i,\vec{e}_i}$ can only have lengths from $\{L + \ell_1 \cdot T^{SCC(s_1)} + \cdots + \ell_k \cdot T^{SCC(s_k)} \mid \ell_1, \ldots, \ell_k \in \mathbb{N}\}$ and, thus, $\{L + \ell \cdot \gcd\{T^{SCC(s_1)}, \ldots, T^{SCC(s_k)}\} \mid \ell \in \mathbb{N}\}$. As $\mathscr{P}(s,t) = \{\pi\}$ in $\mathcal{W}^{s_i,\vec{e}_i}$, $T = \gcd\{T^{SCC(s_1)}, \ldots, T^{SCC(s_k)}\}$. Consequently, all paths from s to t in $\mathcal{W}^{s_i,\vec{e}_i}$ have length in $\{L + \ell T \mid \ell \in \mathbb{N}\}$. Hence, whenever $D^n_{s,t}$ is positive, there are no paths which can contribute positive value to $D^{n+1}_{s,t} + \cdots + D^{n+T-1}_{s,t}$.

For small $n \leq N$, we can always take $c_{s_i,\vec{e}_i}$ small enough so that $D^n_{s,t} \geq c_{s_i,\vec{e}_i}(\rho')^n n^{k'}$ when $D^n_{s,t} > 0$. Take $c'' = \min D^m_{s,t}/(\rho')^m m^{k'}$ where the minimum is over all $m \leq N$ for which $D^m_{s,t} > 0$. Then $D^m_{s,t}/(\rho')^m m^{k'} \geq c''$ for all such m.

This means we can choose $c_{s_i,\vec{e}_i} = \min(c'/2, c'') > 0$, regardless of whether n is big or small: this constant depends on $s_i, \vec{e}_i$ but not on n.

As $c_{s_i,\vec{e}_i}$ depends only on $s_i, \vec{e}_i$, to finish the proof it suffices to take c to be the smallest among the finitely many $c_{s_i,\vec{e}_i}$.

Direction 2 (upper bound). Let $N_{(\rho',k')} = \{n \mid d_{s,t}(n) = (\rho', k')\}$. This gives a finite partition of $\mathbb{N}$ as $\bigcup_{(\rho,k)} N_{(\rho,k)}$. For each (ρ', k'), we shall find a value $C_{(\rho',k')}$ so that, for $n \in N_{(\rho',k')}$, we have $A^n_{s,t} \leq C_{(\rho',k')}(\rho')^n n^{k'}$. Then, to have $A^n_{s,t} \leq C\rho(n)^n n^{k(n)}$ for all $n \in \mathbb{N}$, it will suffice to take C to be the maximum over all $C_{(\rho',k')}$.

Let us fix (ρ', k'). Consider $\mathcal{W}^\bullet$ to be $\mathcal{W}^\dagger$ in which, for every $(\rho, k) \leq (\rho', k')$, we merge the states (t, ρ, k) into a single final state t'. Let us rename the state $(s, 0, 0)$ to s'. (This merger and renaming are justified by assumptions (a) and (b) made at the beginning of this subsection, see p. 24.) Let E be the corresponding transition matrix of $\mathcal{W}^\bullet$. Note that all paths from s' to t' in $\mathcal{W}^\bullet$ go through at most $k' + 1$ SCCs with spectral radius ρ'.

Claim 7.10. *For all* $n \in N_{(\rho',k')}$, *we have* $A^n_{s,t} = E^n_{s',t'}$.

Proof. Consider any path $s \rightarrow q_1 \rightarrow \cdots \rightarrow q_m \rightarrow t$ in $\mathcal{W}$. There is a corresponding path in $\mathcal{W}^\bullet$, however the states q_i are annotated as (q_i, ρ, k), where ρ is the largest spectral radius seen so far, and $k + 1$ is the number of SCC's of that radius number seen so far. The only paths removed are those terminating at (t, ρ, k) with $(\rho, k) > (\rho', k')$. Since $d_{s,t}(n) = (\rho', k')$, we know that no path visits more than $k' + 1$ SCCs of spectral radius ρ', or an SCC of spectral radius greater than ρ'. Consequently, no such path is disallowed in $\mathcal{W}^\bullet$. No paths were added either. Because every SCC in $\mathcal{W}$ remains a strongly connected component in $\mathcal{W}^\bullet$ (duplicated with various (ρ, k)) and its transition probability matrix (and hence the spectral radius) remains the same, we can conclude that $A^n_{s,t} = E^n_{s',t'}$. This completes the proof of Claim 7.10. ∎

Claim 7.11. *There exists* $C_{(\rho',k')}$ *such that* $A^n_{s,t} \leq C_{(\rho',k')}(\rho')^n n^{k'}$.

Proof. We have $A^n_{s,t} = E^n_{s',t'} \leq E^n_{s',t'} + E^{n+1}_{s',t'} + \cdots + E^{n+T(s',t')-1}_{s',t'}$, where $T(s',t')$ is the local period between states s' and t' in $\mathcal{W}^\bullet$. By Theorem 7.5, there exists $C_{(\rho',k')}$ such that this quantity is bounded by $C_{(\rho',k')}(\rho')^n n^{k'}$. Thus, for $n \in N_{(\rho',k')}$, we have $A^n_{s,t} \leq C_{(\rho',k')}(\rho')^n n^{k'}$. This concludes the proof of Claim 7.11. ∎

By the argument above, the upper bound of Claim 7.11 completes the proof of Lemma 7.9. □

Remark 7.12. When $n \leq |Q|$ it is possible that $A_{s,t}^n > 0$ but every path may never take any loops. In which case our degree function allocates $d_{s,t}(n) = (0,0)$, and so Lemma 7.9 would not hold as $A_{s,t}^n \not\leq 0^n n^0$. The behaviour on short words will be handled by the language containment condition.

We will now see how the big-Θ lemma (Lemma 7.9) enables us to characterise the big-O relation on states of weighted automata. For the following lemma, recall the language containment (LC) condition from Definition 6.1 and the ordering on (ρ, k)-pairs from Definition 7.4.

Lemma 7.13. *A state s is big-O of s' if and only if the LC condition holds and, for all but finitely many $n \in \mathbb{N}$, we have $d_{s,t}(n) \leq d_{s',t}(n)$.*

Proof. Let us start off by giving a short summary of the proof. The idea is that, whenever $d_{s,t}(n) \leq d_{s',t}(n)$, by Lemma 7.9, we have $f_s(a^n) \leq (\frac{C}{c}(\frac{\rho}{\rho'})^n n^{k-k'}) \cdot f_{s'}(a^n)$, in which case either $d_{s,t}(n) = d_{s',t}(n)$ and $(\frac{\rho}{\rho'})^n n^{k-k'} = 1$ or $\lim_{n\to\infty} (\frac{\rho}{\rho'})^n n^{k-k'} = 0$ and so $(\frac{\rho}{\rho'})^n n^{k-k'} \leq 1$ for all but finitely many n. At the same time, whenever $d_{s,t}(n) > d_{s',t}(n)$, Lemma 7.9 yields $f_s(a^n) \geq (\frac{c}{C}(\frac{\rho}{\rho'})^n n^{k-k'}) \cdot f_{s'}(a^n)$ but then $\lim_{n\to\infty} (\frac{\rho}{\rho'})^n n^{k-k'} = \infty$.

We now show how to fill in the details in this summary. First we note some consequences of $d_{s,t}(n) \leq d_{s',t}(n)$. Suppose $d_{s,t}(n) = (\rho, k)$ and $d_{s',t}(n) = (\rho', k')$. Thanks to Lemma 7.9, we have $f_s(a^n) \leq (\frac{C}{c}(\frac{\rho}{\rho'})^n n^{k-k'}) \cdot f_{s'}(a^n)$. If $d_{s,t}(n) \leq d_{s',t}(n)$ we can distinguish two cases: either $(\rho, k) = (\rho', k')$ or $(\rho, k) < (\rho', k')$.

- In the former case, $(\frac{\rho}{\rho'})^n n^{k-k'} = 1$ and, thus, $f_s(a^n) \leq (\frac{C}{c}) \cdot f_{s'}(a^n)$.

- In the latter case, we have $\lim_{m\to\infty} (\frac{\rho}{\rho'})^m m^{k-k'} = 0$ and, thus, $(\frac{\rho}{\rho'})^m m^{k-k'} < 1$ for all but finitely many m. Consequently, for all but finitely many n, we can conclude $f_s(a^n) \leq (\frac{C}{c}) \cdot f_{s'}(a^n)$.

Thanks to the above analysis, if $d_{s,t}(n) \leq d_{s',t}(n)$ holds for all but finitely many n, it follows that $f_s(a^n) \leq (\frac{C}{c}) \cdot f_{s'}(a^n)$ for all but finitely many n. Moreover, the language containment condition implies that $f_s(a^n) \leq C' \cdot f_{s'}(a^n)$ for some C' in the remaining (finitely many) cases. Hence, s is big-O of s', which shows the right-to-left implication.

For the converse, recall that we have already established that "s is big-O of s'" implies the language containment condition. For the remaining part, we reason by contrapositive and suppose that there are infinitely many n with $d_{s,t}(n) > d_{s',t}(n)$. As there are finitely many values in the range of $d_{s,t}$ and $d_{s',t}$, there exist (ρ, k) and (ρ', k') such that $(\rho, k) > (\rho', k')$ and, for infinitely many n, $d_{s,t} = (\rho, k)$ and $d_{s',t} = (\rho', k')$. Note that $(\rho', k') \neq (0,0)$, as otherwise $f_{s'}(a^n) = 0$ and, by the language containment condition, $f_s(a^n) = 0$ and $(\rho, k) = (0,0)$, a contradiction. Therefore, $f_{s'}(a^n) > 0$ for all such n, and moreover Lemma 7.9 yields $f_s(a^n) \geq (\frac{c}{C}(\frac{\rho}{\rho'})^n n^{k-k'}) \cdot f_{s'}(a^n)$. But $(\rho, k) > (\rho', k')$ implies

$$\lim_{m\to\infty} \left(\frac{\rho}{\rho'}\right)^m m^{k-k'} = \infty,$$

i.e. $(\frac{\rho}{\rho'})^n n^{k-k'}$ is unbounded. Thus, s cannot be big-O of s'. $\qquad\square$

We are going to use the characterisation from Lemma 7.13 to prove Theorem 7.1. As already discussed, the LC condition can be checked via NFA inclusion testing. To tackle the "for all but finitely many ..." condition, we introduce the concept of eventual inclusion.

Definition 7.14. Given sets A, B, we say A is *eventually included* in B, written $A \subsetneqq B$, if and only if the set difference $A \setminus B$ is finite.

This relation has appeared under the name of "almost inclusion" in, e.g., [Sén93, FT10, IMNN14]. Minimisation of finite-state automata up to finitely many errors has been studied under the name of "hyper-minimization" (see, e.g., [BGS09, Mal11]).

The next three lemmas relate deciding the big-O problem using the characterisation of Lemma 7.13 to eventual inclusion.

Lemma 7.15. *Given unary NFAs $\mathcal{N}_1, \mathcal{N}_2$, the problem $\mathcal{L}(\mathcal{N}_1) \subsetneqq \mathcal{L}(\mathcal{N}_2)$ is in* **coNP**.

Proof of Lemma 7.15. Let M be a DFA accepting $\mathcal{L}(\mathcal{N}_1) \cap \overline{\mathcal{L}(\mathcal{N}_2)}$ obtained through standard automata constructions, i.e. $|M| \leq 2^{|\mathcal{N}_1| + |\mathcal{N}_2|}$. Note that $\mathcal{L}(\mathcal{N}_1) \subsetneqq \mathcal{L}(\mathcal{N}_2)$ if and only if $\mathcal{L}(M)$ is finite. Observe that $\mathcal{L}(M)$ is infinite if and only if there exists $w \in \mathcal{L}(M)$ with $|M| \leq |w| \leq 2|M|$.

Consequently, violation of eventual inclusion can be detected by guessing $n \in \mathbb{N}$ such that $|M| \leq n \leq 2|M|$ and verifying $a^n \in \mathcal{L}(M)$.

Even though M is of exponential size, it is possible to verify $a^n \in \mathcal{L}(M)$ in polynomial time, given n in binary. To this end, we use $\mathcal{N}_1, \mathcal{N}_2$ instead of M and view their transition functions as matrices. Then one can verify the condition using fast matrix exponentiation (by repeated squaring). Because the bit size of n must be polynomial in $|\mathcal{N}_1| + |\mathcal{N}_2|$, the lemma follows. $\square$

Lemma 7.16. *Suppose $d_1, d_2 : \mathbb{N} \to X$, with $(X, \leq)$ a finite total order. Then $d_1(n) \leq d_2(n)$ for all but finitely many n if and only if $\{n \,|\, d_1(n) \geq x\} \subsetneqq \{n \,|\, d_2(n) \geq x\}$ for all $x \in X$.*

Proof. The left-to-right implication is clear. For the opposite direction, observe that, because the order on X is total, $d_1(n) > d_2(n)$ implies the existence of $x \in X$ such that $d_1(n) \geq x$ and $d_2(n) < x$ (it suffices to take $x = d_1(n)$). Because X is finite, $d_1(n) > d_2(n)$ for infinitely many n implies failure of $\{n \,|\, d_1(n) \geq x\} \subsetneqq \{n \,|\, d_2(n) \geq x\}$ for some x. $\square$

Lemma 7.17. *Given a unary weighted automaton $\mathcal{W}$, the associated problem whether $d_{s,t}(n) \leq d_{s',t}(n)$ for all but finitely many $n \in \mathbb{N}$ is in* **coNP**.

Proof. Given an admissible pair $x = (\rho, k)$, we construct an NFA $\mathcal{N}_{s,x}$ accepting $\{a^n \,|\, d_{s,t}(n) \geq x\}$ (similarly $\mathcal{N}_{s',x}$ for s'), by taking the NFA $\mathcal{N}_s(\mathcal{W}^\dagger)$ (Definitions 2.2, 7.8) with a suitable choice of accepting states. Recall that states in $\mathcal{W}^\dagger$ are of the form (q, ρ', k'), where q is a state from $\mathcal{W}$ and (ρ', k') is admissible. If we designate states (t, ρ', k') with $(\rho', k') \geq x$ as accepting, it will accept $\{a^n \,|\, d_{s,t}(n) \geq x\}$. This is a polynomial-time construction.

Then, by Lemma 7.16, the problem whether $d_{s,t}(n) \leq d_{s',t}(n)$ for all but finitely many $n \in \mathbb{N}$ is equivalent to $\mathcal{L}(\mathcal{N}_{s,x}) \subsetneqq \mathcal{L}(\mathcal{N}_{s',x})$ for all admissible x. As there are at most $|Q|^2$ values of x and each can be verified co-nondeterministically in **coNP**, it suffices to show that deciding $\mathcal{L}(\mathcal{N}_{s,x}) \subsetneqq \mathcal{L}(\mathcal{N}_{s',x})$ is in **coNP** for each x. This is the case by Lemma 7.15. $\square$

Lemma 7.13, Remark 6.2, and Lemma 7.17 together complete the upper bound result for Theorem 7.1.

Remark 7.18. Lemma 7.16 may appear simpler using $\{n \,|\, d_1(n) = x\} \subsetneqq \{n \,|\, d_2(n) \geq x\}$. However, it does not seem possible to construct an NFA for $\{a^n \,|\, d_{s,t}(n) = x\}$ in polynomial time. Taking just (t, ρ, k), for $x = (\rho, k)$, as accepting would not be correct, as there could be paths of the same length ending in (t, ρ', k') with $(\rho', k') > (\rho, k)$. Using $d_1(n) \geq x$ instead of $d_1(n) = x$ avoids this problem.

Remark 7.19. An alternative approach for obtaining an upper bound could be to compute the Jordan normal form of the transition matrix and consider its powers. Instead of the interplay of strongly connected components in the transition graph, we would need to consider linear combinations of the nth powers of complex numbers (such as roots of unity). It is not clear this algebraic approach leads to a representation more convenient for our purposes.

7.3. **coNP-hardness for unary LMC.** Given a unary NFA $\mathcal{N}$, the *NFA universality problem* asks if $\mathcal{L}(\mathcal{N}) = \{a^n \mid n \in \mathbb{N}\}$. This problem is **coNP**-complete [SM73]. We exhibit a polynomial-time reduction from (a variant of) the unary universality problem to the big-O problem on unary Markov chains.

Theorem 7.20. *The big-O problem is* **coNP***-hard on unary Markov chains.*

Let us first consider a particular form of unary NFAs.

Definition 7.21. A unary NFA $\mathcal{N} = \langle Q, \rightarrow, q_s, F \rangle$ is in *Chrobak normal form* [Chr86] if
- $Q = S \uplus C^1 \uplus \cdots \uplus C^m$ and $q_s \in S$;
- $S = \{s_1, \cdots, s_k\}$, $q_s = s_1 \in S$ and transitions between states from S form a path $s_1 \xrightarrow{a} s_2 \xrightarrow{a} \ldots \xrightarrow{a} s_k$;
- $C^i = \{c_0^i, \cdots, c_{|C^i|-1}^i\}$ $(1 \le i \le m)$ and transitions between states from C^i form a cycle $c_0^i \xrightarrow{a} c_1^i \xrightarrow{a} \ldots \xrightarrow{a} c_{|C^i|-1}^i \xrightarrow{a} c_0^i$;
- the remaining transitions connect the end of the path to each cycle: $s_k \xrightarrow{a} c_0^i$ for all $1 \le i \le m$.

Any unary NFA can be translated to this representation with at most quadratic blow-up in the size of the machine [Chr86], and such representation can be found in polynomial time [To09, Mar02]. In addition, to simplify our arguments, we introduce a *restricted* Chrobak normal form, which requires that there is exactly one accepting state in each cycle. This restricted form can be found with at most a further quadratic blow-up over Chrobak normal form, by creating copies of cycles—one for each accepting state in the cycle.

Observe that $S \subseteq F$ is a necessary condition for the universality of a unary NFA in Chrobak normal form. Consequently, the universality problem for unary NFA *in restricted Chrobak normal form* such that $k = 1$ is already **coNP**-hard. This is the problem we are going to reduce from in the following.

Proof of Theorem 7.20. Let $\mathcal{N} = \langle Q, \rightarrow, q_s, F \rangle$ be a unary NFA in restricted Chrobak normal form with $k = 1$. We will construct a unary Markov chain $\mathcal{M}$, depicted in Fig. 6, with states $Q' = Q \cup \{s, u, v, t\}$, where t is final. The branch starting from s, defined below, guarantees $f_s(a^n) = \Theta\left(\left(\frac{1}{2}\right)^n\right)$.

$$s \xrightarrow{1} u \qquad u \xrightarrow{\frac{1}{2}} u \qquad u \xrightarrow{\frac{1}{2}} t$$

We take $s' = q_s$ and create a similar branch from s', albeit with a smaller weight, to create paths of weight $\Theta\left(\left(\frac{1}{4}\right)^n\right)$ when reading a^n.

$$s' \xrightarrow{\frac{1}{m+1}} v \qquad v \xrightarrow{\frac{1}{4}} v \qquad v \xrightarrow{\frac{3}{4}} t$$
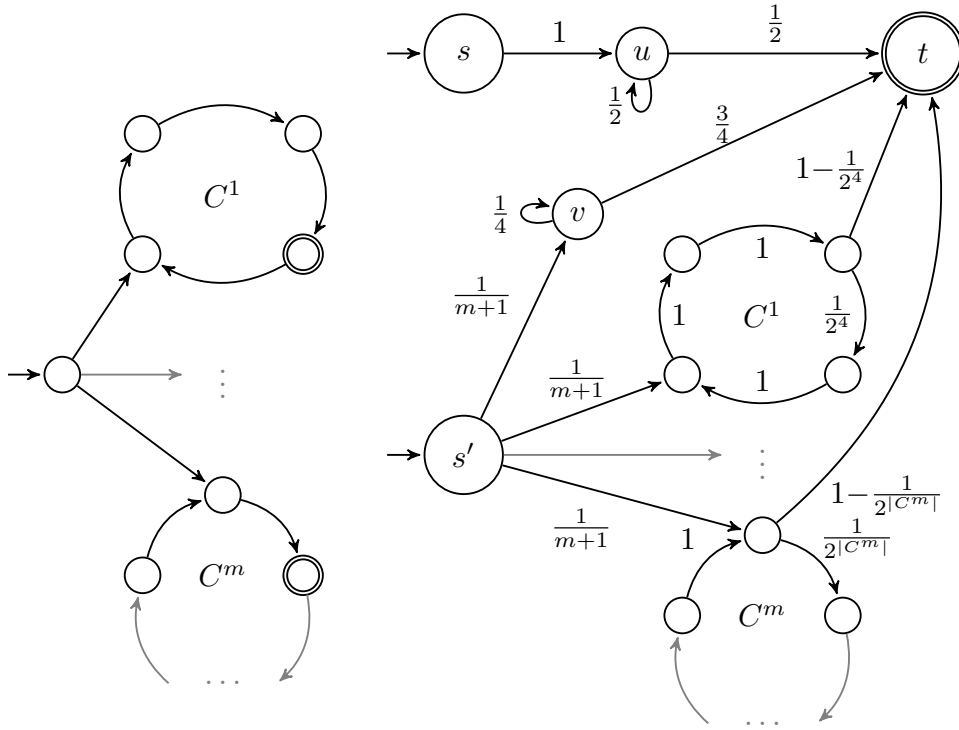
FIGURE 6. Reduction from NFA (left) to LMC (right)

Moreover, we add weights to the original NFA transitions from $\mathcal{N}$ as follows:

$$s' \xrightarrow{\frac{1}{m+1}} c_0^i \quad (1 \leq i \leq m) \qquad c_{j \ominus 1}^i \xrightarrow{(\frac{1}{2})^{|C^i|}} c_j^i \quad (c_j^i \in F)$$

$$c_{j \ominus 1}^i \xrightarrow{1-(\frac{1}{2})^{|C^i|}} t \quad (c_j^i \in F) \qquad c_{j \ominus 1}^i \xrightarrow{1} c_j^i \quad (c_j^i \notin F)$$

where $j \ominus 1 = (|C^i| + j - 1) \bmod |C^i|$. Note that the weights have been selected as if each letter were read with weight $\frac{1}{2}$ except for a bounded number of transitions, where the bound is $\max |C^i|$. Consequently, whenever there are accepting paths for a^n in $\mathcal{N}$, their overall weight in $\mathcal{M}$ will be $\Theta\left((\frac{1}{2})^n\right)$.

It it easy to check that the reduction produces an LMC and can be carried out in polynomial time. It remains to argue that the reduction is correct.

If $\mathcal{N}$ is not universal, there exists n such that $a^n \notin F$. Because of the cyclic structure of Chrobak normal form, $a^{n_k} \notin F$ for $n_k = n + kq$, where $q = \text{lcm}\{|C^1|, \ldots, |C^m|\}$ and $k \in \mathbb{N}$. Then, by the earlier observations about growth, there exists $C > 0$ such that $\sup_k \frac{f_s(a^{n_k})}{f_{s'}(a^{n_k})} = \sup_k C \frac{(1/2)^{n_k}}{(1/4)^{n_k}} = \sup_k C \, 2^{n_k} = \infty$, i.e. s is not big-O of s'.

If $\mathcal{N}$ is universal then, starting from s' in $\mathcal{M}$, every word a^n will have a path weighted $\Theta\left((\frac{1}{4})^n\right)$ as well as paths weighted $\Theta\left((\frac{1}{2})^n\right)$. Hence, there exists $C > 0$ such that

$$\sup_n \left(\frac{f_s(a^n)}{f_{s'}(a^n)} \right) \leq \sup_n \left(C \frac{(\frac{1}{2})^n}{(\frac{1}{4})^n + (\frac{1}{2})^n} \right) \leq C,$$

i.e. s is big-O of s'. □

Remark 7.22. We note that the $\frac{1}{4}$ branch via state v is not strictly necessary, but it demonstrates that the problem is hard even if the LC condition is satisfied (i.e., "it can be the numbers that make the hardness").

8. Decidability for weighted automata with bounded languages

In this section we consider the big-O problem for a weighted automaton $\mathcal{W}$ and states s, s' such that $\mathcal{L}_s(\mathcal{W})$, $\mathcal{L}_{s'}(\mathcal{W})$ are bounded. Throughout the section, we assume that the LC condition has already been checked, i.e. $\mathcal{L}_s(\mathcal{W}) \subseteq \mathcal{L}_{s'}(\mathcal{W})$. We will show that the problem is conditionally decidable, subject to Schanuel's conjecture. We give a quick introduction to this conjecture in Section 8.1.

Theorem 8.1. *Given a weighted automaton* $\mathcal{W} = \langle Q, \Sigma, M, F \rangle$, $s, s' \in Q$, *with* $\mathcal{L}_s(\mathcal{W})$ *and* $\mathcal{L}_{s'}(\mathcal{W})$ *bounded, it is decidable whether* s *is big-O of* s', *subject to Schanuel's conjecture.*

Before delving into proof details, we illustrate the arising challenges using a simple representative example in Section 8.2. After that, the proof of Theorem 8.1 is spread over Sections 8.3 to 8.5. First, a version of Theorem 8.1 restricted to plus-letter-bounded languages is shown (Lemma 8.14 in Section 8.3). Subsequently, letter-bounded languages are also captured by reduction to this case (Lemma 8.15 in Section 8.4). Finally, the case of bounded languages is reduced to the case of letter-bounded languages (Lemma 8.16) in Section 8.5.

8.1. **Logical theories of arithmetic and Schanuel's conjecture.**
In *first-order logical theories of arithmetic,* variables denote numbers (from $\mathbb{Z}$ or $\mathbb{R}$, as appropriate), and atomic predicates are equalities and inequalities between terms built from variables and function symbols. Nullary function symbols are constants, always from $\mathbb{Z}$. If binary addition and multiplication are available, then:

- for $\mathbb{R}$ we obtain the first-order theory of the reals, where the truth value of sentences is decidable due to the celebrated Tarski–Seidenberg theorem [BPR06, Chapter 11 and Theorem 2.77];
- for $\mathbb{Z}$, the first-order theory of the integers is, in contrast, undecidable (see, e.g, [Poo03]).

In the case of $\mathbb{R}$, adding the unary symbol for the exponential function $x \mapsto e^x$ leads to *the first-order theory of the real numbers with exponential function* ($\mathrm{Th}(\mathbb{R}_{\mathsf{exp}})$). Logarithms base 2, for example, are easily expressible in $\mathrm{Th}(\mathbb{R}_{\mathsf{exp}})$. The decidability of $\mathrm{Th}(\mathbb{R}_{\mathsf{exp}})$ is an open problem and hinges upon Schanuel's conjecture [MW96].

Schanuel's conjecture [Lan66] is a unifying conjecture of transcendental number theory, saying that for all $z_1, \ldots, z_n \in \mathbb{C}$ linearly independent over $\mathbb{Q}$ the field extension $\mathbb{Q}(z_1, \ldots, z_n, e^{z_1}, \ldots, e^{z_n})$ has transcendence degree at least n over $\mathbb{Q}$, meaning that for some $S \subseteq \{z_1, \ldots, z_n, e^{z_1}, \ldots, e^{z_n}\}$ of cardinality n, say $S = \{s_1, \ldots, s_n\}$, the only polynomial p over $\mathbb{Q}$ satisfying $p(s_1, \ldots, s_n) = 0$ is $p \equiv 0$. See, e.g., Waldschmidt's book [Wal00, Section 1.4] for further context. If indeed true, this conjecture would generalise several known results, including the Lindemann–Weierstrass theorem and Baker's theorem, and would entail the decidability of $\mathrm{Th}(\mathbb{R}_{\mathsf{exp}})$. A recent exciting line of research reduces problems from verification [DJL$^+$21, MSS20], linear dynamical systems [ACOW18, COW16], and symbolic computation [HLXL18] to the decision problem for $\mathrm{Th}(\mathbb{R}_{\mathsf{exp}})$.
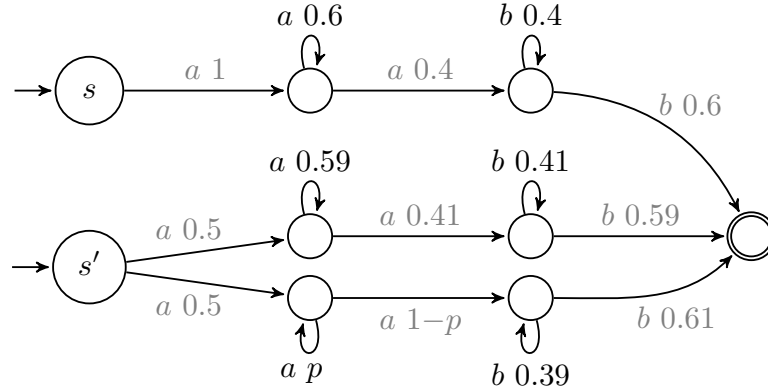
Figure 7. Relative orderings are the same, but the big-O relations are different.

8.2. **Difference to the unary case.** In the unary case, it was sufficient to consider the *relative order* between spectral radii, with careful handling of the periodic behaviour. This approach is insufficient in the bounded case. Example 8.2 highlights that the actual values of the spectral radii have to be examined.

Example 8.2 (Relative orderings are insufficient). Consider the LMC in Fig. 7, with $0.61 \leq p \leq 0.62$. We have $f_s(a^m b^n) = \Theta(0.6^m 0.4^n)$ and $f_{s'}(a^m b^n) = \Theta(p^m 0.39^n + 0.59^m 0.41^n)$. Note that neither $0.59^m 0.41^n$ nor $p^m 0.39^n$ dominate, nor are dominated by, $0.6^m 0.4^n$ for any value of $0.61 \leq p \leq 0.62$. That is, there are values of m, n where $0.59^m 0.41^n \gg 0.6^m 0.4^n$ (in particular large n) and values of m, n where $0.59^m 0.41^n \ll 0.6^m 0.4^n$ (in particular large m); similarly for $p^m 0.39^n$ vs $0.6^m 0.4^n$ (but the cases in which n or m needs to be large are swapped). However, the big-O status can be different for different values of $p \in [0.61, 0.62]$, despite the same relative ordering between spectral radii. When $p = 0.62$, the ratio turns out to be bounded:

$$\frac{f_s(aa^m ab^n b)}{f_{s'}(aa^m ab^n b)} = \frac{1 \cdot 0.6^m \cdot 0.4 \cdot 0.4^n \cdot 0.6}{0.5 \cdot 0.59^m \cdot 0.41 \cdot 0.41^n \cdot 0.59 + 0.5 \cdot 0.62^m \cdot 0.38 \cdot 0.39^n \cdot 0.61} \leq \frac{1600}{1579}$$

for all m, n (in particular, maximal at $m = n = 0$, $\frac{f_s(aab)}{f_{s'}(aab)} = \frac{1600}{1579}$, for all larger m, n the ratio is smaller).

In contrast, when $p = 0.61$, we have $\frac{f_s(a^m b^{0.66m})}{f_{s'}(a^m b^{0.66m})} \xrightarrow{m \to \infty} \infty$. To see that, observe there is an $x \in \mathbb{Q}$ such that $0.61 \cdot 0.39^x < 0.6 \cdot 0.4^x$ and $0.59 \cdot 0.41^x < 0.6 \cdot 0.4^x$, e.g., $x = 0.66$. We can take $m = xn$ such that $m, n \in \mathbb{N}$ and $m, n \to \infty$. Whilst useful for illustration in this example, this effect is not limited to a linear relation between the characters, and so heavier machinery is required.

We first prove Theorem 8.1 for the plus-letter-bounded case, which is the most technically involved; the other bounded cases will be reduced to it. In the plus-letter-bounded case, we will characterise the behaviour of such automata, generalising (ρ, k)-pairs of the unary case. We will need to rely upon the first-order theory of the reals with exponentials to compare these behaviours.

8.3. **The plus-letter-bounded case.** We assume $\mathcal{L}_{s'}(\mathcal{W}) \subseteq a_1^+ \cdots a_m^+$, where $a_1, \cdots, a_n \in \Sigma$ and, because the LC condition holds, we also have $\mathcal{L}_s(\mathcal{W}) \subseteq a_1^+ \cdots a_m^+$. In the plus-letter-bounded cases, without loss of generality, we assume $a_i \neq a_j$ for $i \neq j$. Then any word $w = a_1^{n_1} \ldots a_m^{n_m}$ is uniquely specified by a vector $(n_1, \ldots, n_m) \in \mathbb{N}_{>0}^m$, where n_i is the number of a_i's in w.

Proposition 8.3. *In the plus-letter-bounded cases, without loss of generality, we may assume* $a_i \neq a_j$ *for* $i \neq j$.

Proof. We show how to reduce the big-O problem in the plus-letter-bounded case to the version of the same problem where $a_i \neq a_j$ (for $i \neq j$). Suppose, as above, $\mathcal{L}_{s'}(\mathcal{W}) \subseteq a_1^+ \cdots a_m^+$. We can assume $a_i \neq a_{i+1}$ ($1 \leq i < m$) because $a^+ a^+$ can be replaced with a^+ (even though this results in a less restrictive "bounding" expression). If $\mathcal{W}$ had an a-labelled transition that can be used in two different blocks a_i^+ and a_j^+ ($j \geq i+2$) within $a_1^+ \cdots a_m^+$, then $a = a_i = a_j$ and this transition could be used to "skip" the block a_{i+1}^+, i.e., there would exist a word $w \in \mathcal{L}_{s'}(\mathcal{W})$ that does not use the a_{i+1}^+ block, contradicting $\mathcal{L}_{s'}(\mathcal{W}) \subseteq a_1^+ \cdots a_m^+$. Therefore, every transition can be associated with exactly one block. So we can define a fresh alphabet $\Sigma' = \{b_1, \ldots, b_m\}$ and relabel each transition associated with the ith block by b_i. $\qquad\square$

Our proof plan, for the plus-letter-bounded case (i.e., in the present Section 8.3), will pass the following milestones:

1. Generalise the degree function d from Definition 7.7 to non-unary words. (Definition 8.4 and Definition 8.5 below.)
2. Generalise the big-Θ lemma (Lemma 7.9), to lower- and upper-bound the values $f_s(a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m})$ by more convenient expressions, sometimes called "exponential polynomials". (Lemma 8.6.)
3. Characterise the negative instances of the big-O problem. (Lemma 8.7.)
4. Show that this characterisation is effective, assuming Schanuel's conjecture. (Lemma 8.11.)

A comment on the third milestone is in order. In the unary case, an effective characterisation of positive vis-a-vis negative instances of the big-O problem (Lemma 7.13) can be obtained by "mimicking" the behaviour of the original automaton, once the big-Θ lemma is in place. As it turns out, the non-unary case creates a more subtle, multivariate situation. Our new characterisation (Lemma 8.7) will involve, in particular, the existence of a sequence of vectors with some properties. Accordingly, new arguments will be required (involving further automata and logic) to show algorithmic decidability.

We will now carry out the plan above.

Like in Definition 7.7, we define a degree function d, which will be used to study the asymptotic behaviour of words. This time we will associate a separate (ρ, k) pair to each of the m characters and, consequently, words will induce sequences of the form $(\rho_1, k_1) \cdots (\rho_m, k_m)$.

Further, as there may be multiple, incomparable behaviours, words will induce sets of such sequences, i.e. $d \colon \mathbb{N}^m \to \mathcal{P}((\mathbb{R} \times \mathbb{N})^m)$. For the sake of comparisons, it will be convenient to focus on maximal elements with respect to the pointwise order on $(\mathbb{R} \times \mathbb{N})^m$, written $\leq$, where the lexicographic order (recall Definition 7.4) is used to compare elements of $\mathbb{R} \times \mathbb{N}$.

Recall from the discussion in Remark 7.12, that our big-Θ lemma (Lemma 7.9) does not capture the asymptotics when $n \leq |Q|$ because it cannot distinguish between short paths with no loops and non-paths. In the unary case this is inconsequential as small words are covered by the *finitely many* exceptions and the LC condition. However, here, a small number of one character may be used to enable access to a particular part of the automaton

in another character. For this case, we introduce a new number $\delta = \frac{1}{2} \min_{\varphi : \rho_\varphi > 0} \rho_\varphi$ which is strictly smaller than the spectral radius of every non-zero SCC (so will not dominate with the partial order), but non-zero. This allows us to distinguish between no-path using the $(0,0)$-pair and a short non-looping path using the $(\delta, 0)$-pair.

Definition 8.4. Let $\hat{\rho} = (\rho_1, k_1), \cdots, (\rho_m, k_m) \in (\mathbb{R} \times \mathbb{N})^m$. An $a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m}$-labelled path from s (to the final state) is *compatible with* $\hat{\rho}$ if, for each $i = 1, \ldots, m$, it visits $k_i + 1$ SCCs with spectral radius ρ_i while reading a_i, unless the path visits only singletons with no loops, in which case $(\rho_i, k_i) = (\delta, 0)$. The notation $(\rho, k) \in \hat{\rho}$ is used for "(ρ, k) is an element of $\hat{\rho}$".

Definition 8.5. Let $d_s : \mathbb{N}^m \to \mathcal{P}((\mathbb{R} \times \mathbb{N})^m)$ be s.t.: $\hat{\rho} \in d_s(n_1, \ldots, n_m)$ if and only if

(1) there exists an $a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m}$-labelled path from s to the final state compatible with $\hat{\rho}$, and

(2) for every $a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m}$-labelled path from s compatible with $\hat{\sigma}$ s.t. $\hat{\rho} \leq \hat{\sigma}$, we have $\hat{\rho} = \hat{\sigma}$.

Observe that $\hat{\rho}$ may range over at most $|Q|^{2m}$ possible values. Let $\mathcal{D}$ be the set of all of them, so that $d_s : \mathbb{N}^m \to \mathcal{P}(\mathcal{D})$. In this extended setting, the big-Θ lemma (Lemma 7.9) may be generalised as follows.

Lemma 8.6. Denote $z(n_1, \cdots, n_m) = \sum_{\hat{\rho} \in d_s(n_1, \ldots, n_m)} \prod_{(\rho_i, k_i) \in \hat{\rho}} \rho_i^{n_i} \cdot n_i^{k_i}$.
There exist $c, C > 0$ such that for all $n_1 \ldots, n_m \in \mathbb{N}$:

$$c \cdot z(n_1, \cdots, n_m) \leq f_s(a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m}) \leq C \cdot z(n_1, \cdots, n_m).$$

Proof.

$$\begin{aligned}
f_s(a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m}) &= (M(a_1)^{n_1} \times M(a_2)^{n_2} \times \cdots \times M(a_m)^{n_m})_{s,t} \\
&= \sum_{q_1 \in Q} M(a_1)^{n_1}_{s,q_1} (\times M(a_2)^{n_2} \times \cdots \times M(a_m)^{n_m})_{q_1, t} \\
&\vdots \\
&= \sum_{(q_1, \ldots, q_{m-1}) \in Q^{m-1}} M(a_1)^{n_1}_{s,q_1} \times M(a_2)^{n_2}_{q_1, q_2} \times \cdots \times M(a_m)^{n_m}_{q_{m-1}, t}
\end{aligned}$$

Let $s = q_0$ and $t = q_m$, then a sequence $q_0, \ldots, q_m$ describes a possible path through the automaton.

By Lemma 7.9 in the unary case, for each $M(a_i)^{n_i}_{q_{i-1}, q_i} > 0$, there is a $(\rho_{q_{i-1}, q_i}, k_{q_{i-1}, q_i}), c, C$, such that if $n_i > |Q|$

$$c \rho_{q_{i-1}, q_i}^{n_i} n_i^{k_{q_{i-1}, q_i}} \leq M(a_i)^{n_i}_{q_{i-1}, q_i} \leq C \rho_{q_{i-1}, q_i}^{n_i} n_i^{k_{q_{i-1}, q_i}}. \tag{8.1}$$

Otherwise if $n_i \leq |Q|$, there exists c, C satisfying Eq. (8.1) regardless of the value of $d_{q_{i-1}, q_i}(n_i) = (\rho, k)$, since there are at most $|Q|$ characters thus bounding $M(a_i)^{n_i}_{q_{i-1}, q_i}$. In particular Eq. (8.1) holds for $(\rho, k) = (\delta, 0)$, that is, if no path of length n_i between q_{i-1} and q_i goes through a loop.

When $M(a_i)^{n_i}_{q_{i-1}, q_i}$ is zero, $M(a_i)^{n_i}_{q_{i-1}, q_i}$ correspond to the $(0,0)$-pair, and $M(a_i)^{n_i}_{q_{i-1}, q_i} = 0^{n_i} n_i^0 = 0$).

Take c, C so that C is maximised over all such C and c is minimised over all such c. We have $c^m \tilde{z}(n_1, \ldots, n_m) \leq f_s(a_1^{n_1} a_2^{n_2} \ldots a_m^{n_m}) \leq C^m \tilde{z}(n_1, \ldots, n_m)$, where

$$\tilde{z}(n_1, \ldots, n_m) = \sum_{(q_1, \ldots, q_{m-1}) \in Q^{m-1}} \rho_{s,q_1}^{n_1} n_1^{k_{s,q_1}} \cdot \ldots \cdot \rho_{q_{m-1},t}^{n_m} n_m^{k_{q_{m-1},t}}.$$

We observe that $\tilde{z}(n_1, \ldots, n_m)$ is nearly $z(n_1, \ldots, n_m)$. First we omit paths which contain a $(0,0)$ element, then each summand $\rho_{s,q_1}^{n_1} n_1^{k_{s,q_1}} \cdot \ldots \cdot \rho_{q_{m-1},t}^{n_m} n_m^{k_{q_{m-1},t}}$ of the sum corresponds with a candidate element $\hat{\rho}$ of $d_s(n_1, \ldots, n_m)$. Given any two elements where $\hat{\rho} = \hat{\sigma}$ only one representative need be kept in $d_s(n_1, \ldots, n_m)$ by replacing c and C with $c/2$ and $2C$.

Similarly, we only need to keep maximal elements. By standard manipulations, given $\hat{\sigma} = (\sigma_1, \ell_1) \ldots (\sigma_m, \ell_m), \hat{\rho} = (\rho_1, k_1) \ldots (\rho_m, k_m)$, such that $(\sigma_i, \ell_i) \leq (\rho_i, k_i)$ for all $i \in \{1, \ldots, m\}$, then for some c', C' we have

$$c' \prod_{(\rho_i, k_i) \in \hat{\rho}} \rho_i^{n_i} n_i^{k_i} \leq \prod_{(\sigma_i, \ell_i) \in \hat{\sigma}} \sigma_i^{n_i} n_i^{\ell_i} + \prod_{(\rho_i, k_i) \in \hat{\rho}} \rho_i^{n_i} n_i^{k_i} \leq C' \prod_{(\rho_i, k_i) \in \hat{\rho}} \rho_i^{n_i} n_i^{k_i}.$$

and so $\hat{\sigma}$ can be omitted from $d_s(n_1, \ldots, n_m)$ by replacing c and C with cc' and CC'. $\square$

The following lemma provides the key characterisation of negative instances of the big-O problem, in the plus-letter-bounded case and assuming the LC condition. Here and below, we write $n(t)$ to refer to the the tth vector in a sequence $n \colon \mathbb{N} \to \mathbb{N}^m$.

Lemma 8.7 (Main lemma). *Assume $\mathcal{L}_s(\mathcal{W}) \subseteq \mathcal{L}_{s'}(\mathcal{W})$. Then s is not big-O of s' if and only if there exists a sequence $n \colon \mathbb{N} \to \mathbb{N}^m$ and $X \in \mathcal{D}, \mathcal{Y} \subseteq \mathcal{D}$ such that*

(a) *$X \in d_s(n(t))$ and $\mathcal{Y} = d_{s'}(n(t))$ for all t, and*
(b) *for all $j \in h_\mathcal{Y}$, the sequence n satisfies*

$$\sum_{i=1}^{m} \alpha_{j,i} \, n(t)_i + p_{j,i} \log n(t)_i \xrightarrow[t \to \infty]{} -\infty,$$

where $h_\mathcal{Y} \subseteq \{1, \ldots, |\mathcal{Y}|\}$, $\alpha_{j,i} \in \mathbb{R}$, $p_{j,i} \in \mathbb{Z}$ ($1 \leq i \leq m$) are uniquely determined by X and $\mathcal{Y}$ (in a way detailed below), $h_\mathcal{Y}$ and $p_{j,i}$'s are effectively computable and $\alpha_{j,i}$'s are first-order expressible (with exponential function).

Proof. Observe that then s is *not* big-O of s' iff there exists an infinite sequence of words such that, for all $C > 0$, the sequence contains a word w such that $\frac{f_s(w)}{f_{s'}(w)} > C$. Thanks to Lemma 8.6, this is equivalent to the existence of a sequence $n \colon \mathbb{N} \to \mathbb{N}^m$ such that

$$\frac{\displaystyle\sum_{X \in d_s(n(t)_1, \ldots, n(t)_m)} \prod_{(\rho_i, k_i) \in X} \rho_i^{n(t)_i} n(t)_i^{k_i}}{\displaystyle\sum_{Y \in d_{s'}(n(t)_1, \ldots, n(t)_m)} \prod_{(\sigma_i, \ell_i) \in Y} \sigma_i^{n(t)_i} n(t)_i^{\ell_i}} \xrightarrow[t \to \infty]{} \infty,$$

where $n(t)_i$ denotes the ith component of $n(t)$. Since there are finitely many possible values of d_s and $d_{s'}$, it suffices to look for sequences n such that $d_s(n(t))$ and $d_{s'}(n(t))$ are fixed. Further, because of the sum in the numerator, only one $X \in \mathcal{X}$ is required such that $X \in d_s(n_1, \ldots, n_m)$. Thus, we need to determine whether there exist $X \in \mathcal{D}, \mathcal{Y} \subseteq \mathcal{D}$ and $n \colon \mathbb{N} \to \mathbb{N}^m$ such that $X \in d_s(n(t))$, $d_{s'}(n(t)) = \mathcal{Y}$ (for all t) and

$$\frac{\prod_{i=1}^{m} \rho_i^{n(t)_i} n(t)_i^{k_i}}{\sum_{j=1}^{h_\mathcal{Y}} \prod_{i=1}^{m} \sigma_{ji}^{n(t)_i} n(t)_i^{\ell_{ji}}} \xrightarrow[t \to \infty]{} \infty.$$

where $X = (\rho_1, k_1) \cdots (\rho_m, k_m)$, $\mathcal{Y} = \{Y_1, \cdots, Y_{|\mathcal{Y}|}\}$, and $Y_j = (\sigma_{j1}, \ell_{j1}) \cdots (\sigma_{jm}, \ell_{jm})$ $(1 \leq j \leq |\mathcal{Y}|)$. Taking the reciprocal and requiring each of the summands to go to zero, we obtain

$$\frac{\prod_{i=1}^{m} \sigma_{ji}^{n(t)_i} \, n(t)_i^{\ell_{ji}}}{\prod_{i=1}^{m} \rho_i^{n(t)_i} \, n(t)_i^{k_i}} = \prod_{i=1}^{m} \left(\frac{\sigma_{ji}}{\rho_i}\right)^{n(t)_i} n(t)_i^{\ell_{ji}-k_i} \xrightarrow[t\to\infty]{} 0 \qquad \text{for all } 1 \leq j \leq |\mathcal{Y}|.$$

If we take logarithms, letting $\alpha_{j,i} = \log(\frac{\sigma_{ji}}{\rho_i})$ and $p_{j,i} = \ell_{ji} - k_i$, we get

$$\sum_{i=1}^{m} \alpha_{j,i} \, n(t)_i + p_{j,i} \log n(t)_i \xrightarrow[t\to\infty]{} -\infty$$

for all j in $h_\mathcal{Y} = \{1 \leq j \leq |\mathcal{Y}| \mid \sigma_{ji} > 0 \text{ for all } 1 \leq i \leq m\}$.

The number $\alpha_{j,i}$ is the logarithm of the ratio of two algebraic numbers, which are not given explicitly. However, they admit an unambiguous, first-order expressible characterisation (recall Lemma 7.3). The logarithm is encoded using the exponential function: $\log(z)$ is the $x \in \mathbb{R}$ such that $\exp(x) = z$. $\qquad\square$

Lemma 8.7 identifies violation of the big-O property using two conditions. In the remainder of this subsection we will handle Condition (a) using automata-theoretic tools (the Parikh theorem and semi-linear sets) and Condition (b) using logics. In summary, the characterisation of Lemma 8.7 will be expressed in the first-order theory of the reals with exponentiation, which is decidable subject to Schanuel's conjecture.

Condition (a) via automata. It turns out that sequences n satisfying Condition (a) in Lemma 8.7 can be captured by a finite automaton. In more detail, for any $X \in \mathcal{D}$, there exists an automaton $\mathcal{N}_X^s$ such that $\mathcal{L}(\mathcal{N}_X^s) = \{a_1^{n_1} \cdots a_m^{n_m} \mid X \in d_s(n_1, \cdots, n_m)\}$. For any $\mathcal{Y} \subseteq \mathcal{D}$, there exists an automaton $\mathcal{N}_\mathcal{Y}^s$ such that $\mathcal{L}(\mathcal{N}_\mathcal{Y}^s) = \{a_1^{n_1} \cdots a_m^{n_m} \mid d_s(n_1, \cdots, n_m) = \mathcal{Y}\}$. The relevant automaton capturing X and $\mathcal{Y}$ is then found by taking the intersection of $\mathcal{L}(\mathcal{N}_X^s)$ and $\mathcal{L}(\mathcal{N}_\mathcal{Y}^{s'})$.

Lemma 8.8. *For any $X \in \mathcal{D}$ and $\mathcal{Y} \subseteq \mathcal{D}$, there exists an automaton $\mathcal{N}_{X,\mathcal{Y}}$ such that $\mathcal{L}(\mathcal{N}_{X,\mathcal{Y}}) = \{a_1^{n_1} \cdots a_m^{n_m} \mid X \in d_s(n_1, \cdots, n_m), \mathcal{Y} = d_{s'}(n_1, \cdots, n_m)\}$.*

Proof. Let $\hat{\rho} = (\rho_1, k_1) \cdots (\rho_m, k_m)$. Building on Definition 7.8, one can construct an automaton $\mathcal{N}_{\geq\hat{\rho}}^s$ with

$$\mathcal{L}(\mathcal{N}_{\geq\hat{\rho}}^s) = \{a_1^{n_1} \ldots a_m^{n_m} \mid \exists \hat{\sigma} \in d_s(n_1, \ldots, n_m) \; \hat{\sigma} \geq \hat{\rho}\}$$

by tracking the current maximum spectral radius seen and the number of different SCCs with this spectral radius. If the only states seen so far have been singletons with no loops (formally having spectral radius 0), the value should be tracked as $(\delta, 0)$ regardless of how many have been seen.

Passage from states reading a_j to states reading a_{j+1} is allowed only if the tracked value is at least (ρ_j, k_j), and states should be final if the tracked value of a_m is at least (ρ_m, k_m). As previously, comparisons are with respect to the partial order of Definition 7.4.

Similarly, one can construct $\mathcal{N}_{>\hat{\rho}}^s$ with

$$\mathcal{L}(\mathcal{N}_{>\hat{\rho}}^s) = \{a_1^{n_1} \ldots a_m^{n_m} \mid \exists \hat{\sigma} \in d_s(n_1, \ldots, n_m) \; \hat{\sigma} > \hat{\rho}\}.$$

The construction is the same as for $\mathcal{N}_{\geq\hat{\rho}}^s$ except that, in order to accept, we need to be sure that at least one of the 'at least' comparisons was strict. This can be achieved by maintaining an extra bit at run time.

Note that $\mathcal{L}(\mathcal{N}^s_{\geq\hat\rho}) \backslash \mathcal{N}^s_{>\hat\rho}$ contains all $a_1^{n_1} \ldots a_m^{n_m}$ such that there exists $\hat\sigma \in d_s(n_1, \cdots, n_m)$ with $\hat\sigma \geq \hat\rho$ and, for all $\hat\tau \in d_s(n_1, \cdots, n_m)$, we do *not* have $\hat\tau > \hat\rho$. Consequently, we must have $\hat\rho \in d_s(n_1, \cdots, n_m)$, which implies (by maximality) that we cannot have $\hat\tau > \hat\rho$ for any $\hat\sigma \in d_s(n_1, \cdots, n_m)$. Hence,

$$\mathcal{L}(\mathcal{N}^s_{\geq\hat\rho}) \setminus \mathcal{L}(\mathcal{N}^s_{>\hat\rho}) = \{a_1^{n_1} \ldots a_m^{n_m} \mid \hat\rho \in d_s(n_1, \cdots, n_m)\}.$$

Denote by $\mathcal{N}^s_{\hat\rho}$ the corresponding automaton; we can take $\mathcal{N}^s_X = \mathcal{N}^s_{\hat\rho}$.

Given $\mathcal{Y} \subseteq \mathcal{D}$, we can then take $\mathcal{N}^{s'}_{\mathcal{Y}}$ to be the automaton corresponding to

$$\bigcap_{\hat\rho\in\mathcal{Y}} \mathcal{L}(\mathcal{N}^{s'}_{\hat\rho}) \cap \bigcap_{\hat\rho\in\mathcal{D}\backslash\mathcal{Y}} \left(a_1^+ \cdots a_m^+ \setminus \mathcal{L}(\mathcal{N}^{s'}_{\hat\rho})\right).$$

The relevant automaton $\mathcal{N}_{X,\mathcal{Y}}$ is then found by taking the intersection of $\mathcal{L}(\mathcal{N}^s_X)$ and $\mathcal{L}(\mathcal{N}^{s'}_{\mathcal{Y}})$. $\square$

Because of our $a_i \neq a_j$ assumption, the vector $(n_1, \cdots, n_m)$ indicates the number of occurrences of each character. The set of such vectors derived from the language of an automaton is known as the *Parikh image* of this language [Par66]. It is well known that the Parikh image of every regular language is a semi-linear set, i.e. a finite union of linear sets (a linear set has the form $\{\vec{b} + \lambda_1\vec{r}^1 + \cdots + \lambda_s\vec{r}^s \mid \lambda_1, \ldots, \lambda_s \in \mathbb{N}\}$, where $\vec{b} \in \mathbb{N}^m$ is the base vector and $\vec{r}^1, \cdots \vec{r}^s \in \mathbb{N}^m$ are called period vectors). However, since $\mathcal{L}(\mathcal{N}_{X,\mathcal{Y}}) \subseteq a_1^+ a_2^+ \ldots a_m^+$, the linear sets are of a very particular form, where each $\vec{r}^i$ is a constant multiple of the ith unit vector.

Lemma 8.9. *The Parikh image of every plus-letter-bounded regular language is a finite union of linear sets, where each period vector is a constant multiple of some unit vector.*

In particular, the language of $\mathcal{N}_{X,\mathcal{Y}}$ can be effectively decomposed as $\mathcal{L}(\mathcal{N}_{X,\mathcal{Y}}) = \bigcup_{k=1}^{S_{X,\mathcal{Y}}} \mathcal{L}_k$, where $\mathcal{L}_k = \left\{a_1^{b_{k1}+r_{k1}\lambda_1} \cdots a_m^{b_{km}+r_{km}\lambda_m} \mid \lambda_1, \cdots, \lambda_m \in \mathbb{N}\right\}$, $S_{X,\mathcal{Y}} \in \mathbb{N}$ and $b_{ki}, r_{ki} \in \mathbb{N}$ ($1 \leq k \leq S_{X,\mathcal{Y}}$, $1 \leq i \leq m$).

Proof. Consider the machine $\mathcal{N}_{X,\mathcal{Y}}$, accepting a language which is a subset of $a_1^+ a_2^+ \ldots a_m^+$, with any state not reachable from the starting state or not leading to an accepting state removed. To induce a form with the property we want, we intersect $\mathcal{N}_{X,\mathcal{Y}}$ with the standard DFA[4] for $a_1^+ a_2^+ \ldots a_m^+$, without changing the language.

Hence every state corresponds to reading from exactly one character block of $a_1, a_2, \ldots, a_m$. At each state there can be at most two characters enabled, either the character to remain in the current character block, or the character to move to the next. Every state can be labelled as

- *only having transition for a_i*; or
- *also having transition with a_{i+1}.*

Consider all possible choices of automaton formed by restricting $\mathcal{N}_{X,\mathcal{Y}}$ so that there is a single state which is allowed to transition from a_i to a_{i+1} for each i and any other state which had this property in $\mathcal{N}_{X,\mathcal{Y}}$ has its a_{i+1} transitions removed (but keeps its a_i transitions). Each such choice corresponds with a partition of the accepting runs of $\mathcal{N}_{X,\mathcal{Y}}$.

---

[4]By DFA we permit a partial transition function, that is 0 or 1 transition for each character from every state, rather than exactly 1.

Thus $\mathcal{L}(\mathcal{N}_{X,\mathcal{Y}})$ is the finite union over the languages induced by all such machines. We further show that such machines can further be expressed as a finite union of linear sets in the form prescribed.

Let us assume $\mathcal{N}_{X,\mathcal{Y}}^j$ is such a machine with a single state capable of transitioning from a_i to a_{i+1} for each i, and again remove any state not reachable from the starting state or not leading to an accepting state. The part of the machine reading a_i has a single starting state and a single final state, which is a unary NFA when the transitions to a_{i+1} are discarded.

This unary NFA can be converted to Chrobak normal form; the section of $\mathcal{N}_{X,\mathcal{Y}}^j$ corresponding to a_i can be replaced with this unary NFA, and any accepting state has additionally the transitions for transitioning from a_i to a_{i+1} of the single such state in $\mathcal{N}_{X,\mathcal{Y}}^j$.

Let us repeat the process above for all i, decomposing $\mathcal{N}_{X,\mathcal{Y}}^j$ into the subsets of languages where there are exactly one state transitioning from a_i to a_{i+1}. Let $\mathcal{N}_{X,\mathcal{Y}}^j = \bigcup_k \mathcal{N}_{X,\mathcal{Y}}^{j,k}$, a finite union; where each k corresponds to a selection of accepting states $(q_1, \dots, q_m)$ with q_l being the accepting state in the Chrobak normal form for a_l.

Consider such an $\mathcal{N}_{X,\mathcal{Y}}^{j,k}$. The steps spent in each block corresponding to a_i is either formed by the finite path or the a single cycle at the end of the path. If the transition occurs in the finite path then b_{ki} is the length of the path to that transition and r_{ki} is zero. If the transition occurs in the cycle at the end of the path, then b_{ki} is the length of the path to that transition from the start of the path and r_{ki} is the length of the cycle. In $\mathcal{N}_{X,\mathcal{Y}}^{j,k}$ the time spent in block a_i has no influence on the time spent in a_j for $j \neq i$. Then $\mathcal{L}(\mathcal{N}_{X,\mathcal{Y}}^{j,k}) = \{a^{n_1} a^{n_2} \dots a^{n_m} | \exists \vec{\lambda} \in \mathbb{N}^m \text{ s.t. } \forall i \in [m] n_i = b_{ki} + r_{ki} \cdot \lambda_i\}$. The language $\mathcal{L}(\mathcal{N}_{X,\mathcal{Y}})$ is the union over all $\mathcal{L}(\mathcal{N}_{X,\mathcal{Y}}^{j,k})$. $\qquad\square$

Lemma 8.9 captures Condition (a) of Lemma 8.7 precisely.

Condition (b) via logic. With Lemma 8.9 in place, we now move on to add Condition (b) to the existing machinery. In fact, the logical formulae in the following lemmas will express the conjunction of both conditions of Lemma 8.7.

Lemma 8.10. *Assume $\mathcal{L}_s(\mathcal{W}) \subseteq \mathcal{L}_{s'}(\mathcal{W})$. Then s is not big-O of s' if and only if there exists $X \in \mathcal{D}$, $\mathcal{Y} \subseteq \mathcal{D}$, $1 \leq k \leq S_{X,\mathcal{Y}}$ such that*

$$\forall C < 0 \; \exists \vec{\lambda} \in \mathbb{N}^m \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i=1}^m \alpha_{j,i} (b_{ki} + r_{ki} \lambda_i) + p_{j,i} \log(b_{ki} + r_{ki} \lambda_i) < C,$$

where $h_{\mathcal{Y}}, \alpha_{j,i}, p_{j,i}$ (resp. b_{ki}, r_{ki}) satisfy the same conditions as in Lemma 8.7 (resp. 8.9).

Note that the formula of Lemma 8.10 uses quantification over natural numbers. Our next step will be to replace integer variables with real variables. In other words, we will obtain an equivalent condition in the first-order theory of the reals with exponentiation, as follows.

Lemma 8.11. *Assume $\mathcal{L}_s(\mathcal{W}) \subseteq \mathcal{L}_{s'}(\mathcal{W})$. Then s is not big-O of s' if and only if there exist $X \in \mathcal{D}$, $\mathcal{Y} \subseteq \mathcal{D}$, $1 \leq k \leq S_{X,\mathcal{Y}}$ and $U \subseteq \{i \in \{1, \cdots, m\} \mid r_{ki} > 0\}$ such that*

$$\forall C < 0 \; \exists \vec{x} \in \mathbb{R}_{\geq B_k}^{|U|} \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i \in U} \alpha_{j,i} r_{ki} x_i + p_{j,i} \log(x_i) < C,$$

where $B_k = \max_i b_{ki}$ and $h_{\mathcal{Y}}, \alpha_{j,i}, p_{j,i}, b_{ki}, r_{ki}$ are as in Lemma 8.10.

Proof idea. Compare the logical characterisation in Lemma 8.11 and Lemma 8.10. The first difference to note is that the effect of b_{ki}'s is simply a constant offset, and so the sequence would tend to $-\infty$ with or without its presence. Similar simplifications can be made inside the logarithm: the multiplicative effect of r_{ki} inside the logarithm can be extracted as an additive offset and thus similarly be discarded.

The second crucial difference is to relax the variable domains from integers to reals. If each of the λ_i in the satisfying assignment is sufficiently large, we show we can relax the condition to real numbers rather than integers without affecting whether the sequence goes to $-\infty$. To do this, we test sets of indices U, where if $i \in U$ then λ_i needs to be arbitrarily large over all C (i.e. unbounded). The positions where λ_i is always bounded are again a constant offset and are omitted. $\qquad\square$

Proof of Lemma 8.11. First we argue that we can restrict to some subset of the components which enable the satisfying choice of λ to be sufficiently large in all components.

Claim 8.12. *The assertion*

$$\forall C \; \exists \vec{\lambda} \in \mathbb{N}^m \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i=1}^{m} \alpha_{j,i} \left(b_{ki} + r_{ki}\,\lambda_i\right) + p_{j,i} \log(b_{ki} + r_{ki}\,\lambda_i) < C \tag{8.2}$$

holds if and only if the following assertion holds for some $U \subseteq [m]$:

$$\forall C \; \exists \vec{\lambda} \in \mathbb{N}^U_{\geq \max_i b_{ki}} \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i \in U} \alpha_{j,i} \left(b_{ki} + r_{ki}\,\lambda_i\right) + p_{j,i} \log(b_{ki} + r_{ki}\,\lambda_i) < C. \tag{8.3}$$

Proof of Claim 8.12. First note that Eq. (8.3) immediately implies Eq. (8.2). We show the converse.

Recall we can alternatively characterise the formulation as a sequence $n : \mathbb{N} \to \mathbb{N}^m$. That is, for each negative integer C, the choice of $\vec{\lambda}$ corresponds to $n(C)$ in the sequence.

Note that in the sequence n some components may be bounded. Either because $r_{ki} = 0$, or the choice of n makes it so. Suppose there exists a $\theta > 0$ such that $n(t)_x \leq \theta$ for some $x \in [m]$, then $\sum_{i=1}^{m} \alpha_{j,i}\, n(t)_i + p_{j,i} \log(n(t)_i) \leq \sum_{i=1, i \neq x}^{m} \alpha_{j,i}\, n(t)_i + p_{j,i} \log(n(t)_i) + |\alpha_{j,i}|\,\theta + |p_{j,i}|\theta$. Hence the sequence $\sum_{i=1, i \neq x}^{m} \alpha_{j,i}\, n(t)_i + p_{j,i} \log(n(t)_i)$ goes to $-\infty$ as well.

Consider each choice of components $B \subseteq [m]$ which will be bounded. For some components there will be no choice as $r_{ki} = 0$. Let us assume that the chosen set is maximal with respect to set-inclusion; that is, there should be no subsequence maintaining the property with fewer components unbounded. Let the remaining unbounded components be $U = [m] \setminus B$.

Since each remaining component is not bounded, there is always a later point in the sequence in which the value is larger; thus one can take a subsequence of $n(t)$ so that $n(t)_i \leq n(t+1)_i$ for every t. Repeat for every remaining component $i \in U$; this can be done as the minimal choice of unbounded components has been selected. Hence, without loss of generality if there exists some sequence, then for any θ, there exists a subsequence of $n(t)$, such that $n(t)_i > \theta$ for all $i \in U$. To enable a more succinct analysis later, restrict $n(t)$ to those in which $\lambda_i \geq \max_i b_{ki}$ where $n(t)_i = b_{ki} + r_{ki}\,\lambda_i$ for some λ_i. $\qquad\blacksquare$

Next we argue that the offset component $\vec{b}$ does not affect whether the formula holds and that we can relax the restriction of $\vec{\lambda}$ from naturals to positive reals and maintain the satisfiability of the formula. The advantage here is that this relaxation can be solved with the first order theory of the reals with exponential function; which is decidable subject to Schanuel's conjecture.

Claim 8.13. *The assertion*

$$\forall C \; \exists \vec{\lambda} \in \mathbb{N}^U_{\geq \max_i b_{ki}} \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i \in U} \alpha_{j,i} \left(b_{ki} + r_{ki} \lambda_i \right) + p_{j,i} \log(b_{ki} + r_{ki} \lambda_i) < C \qquad (8.4)$$

holds if and only if the following assertion holds:

$$\forall C \; \exists \vec{x} \in \mathbb{R}^U_{\geq \max_i b_{ki}} \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i \in U} \alpha_{j,i} r_{ki} x_i + \sum_{i \in U} p_{j,i} \log(x_i) < C. \qquad (8.5)$$

Proof of Claim 8.13. Observe that

$$\sum_{i \in U} \alpha_{j,i} \left(b_{ki} + r_{ki} \lambda_i \right) = \sum_{i \in U} \alpha_{j,i} b_{ki} + \sum_{i \in U} \alpha_{j,i} r_{ki} \lambda_i$$

and that $\sum_{i \in U} \alpha_{j,i} b_{ki}$ is constant so it does not affect whether the sequence goes to $-\infty$, hence Eq. (8.4) holds if and only if

$$\forall C \; \exists \vec{\lambda} \in \mathbb{N}^U_{\geq \max_i b_{ki}} \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i \in U} \alpha_{j,i} r_{ki} \lambda_i + p_{j,i} \log(b_{ki} + r_{ki} \lambda_i) < C \qquad (8.6)$$

Now let us extract the log component by using the following rewriting

$$\log(b_{ki} + r_{ki} \lambda_i) = \log\left(\lambda_i \left(\frac{b_{ki}}{\lambda_i} + r_{ki} \right) \right) = \log(\lambda_i) + \log\left(\frac{b_{ki}}{\lambda_i} + r_{ki} \right).$$

Since $r_{ki} \geq 1$ and $\lambda_i \geq b_{ki}$ we have $\log\left(\frac{b_{ki}}{\lambda_i} + r_{ki} \right) \leq \log(r_{ki} + 1)$, which is constant. Hence Eq. (8.4) is equivalent to:

$$\forall C' \; \exists \vec{\lambda} \in \mathbb{N}^U_{\geq \max_i b_{ki}} \bigwedge_{j \in h_{\mathcal{Y}}} \sum_{i \in U} \alpha_{j,i} r_{ki} \lambda_i + \sum_{i \in U} p_{j,i} \log(\lambda_i) < C' \qquad (8.7)$$

We now show that this is equivalent to Eq. (8.5). Clearly Eq. (8.7) implies Eq. (8.5). Now consider Eq. (8.5) holding, and we show the Eq. (8.7) is satisfied, by exhibiting a choice of $\vec{\lambda} \in \mathbb{N}^U_{\geq \max_i b_{ki}}$ for every C'. Given $C' < 0$, let $C = C' - \max_j \sum_{i \in U} |\alpha_{j,i}| r_{ki} - \max_j \sum_{i \in U} |p_{j,i}|$, and choose $\vec{x} \in \mathbb{R}^{|U|}_{\geq \max_i b_{ki}}$ satisfying Eq. (8.5). Now let $x_i = \lambda_i + y_i$, with $y_i < 1, \lambda_i = \lfloor x_i \rfloor$. First observe that since $x_i \geq \max_i b_{ki}$, an integer, also $\lambda_i \geq \max_i b_{ki}$. Observe that $\left| \sum_{i \in U} \alpha_{j,i} r_{ki} y_i \right| \leq \sum_{i \in U} |\alpha_{j,i}| r_{ki}$. Since

$$\sum_{i \in U} \alpha_{j,i} r_{ki} \lambda_i + \sum_{i \in U} \alpha_{j,i} r_{ki} y_i + \sum_{i \in U} p_{j,i} \log(\lambda_i + y_i) < C$$

we have

$$\sum_{i \in U} \alpha_{j,i} r_{ki} \lambda_i + \sum_{i \in U} p_{j,i} \log(\lambda_i + y_i) < C + \sum_{i \in U} |\alpha_{j,i}| r_{ki}$$

Let us again rewrite $\log(\lambda_i + y_i) = \log\left(\lambda_i \left(1 + \frac{y_i}{\lambda_i} \right) \right) = \log(\lambda_i) + \log\left(1 + \frac{y_i}{\lambda_i} \right)$. Then since $\lambda_i > y_i$, $\log\left(1 + \frac{y_i}{\lambda_i} \right) \leq 1$, so

$$\left| \sum_{i \in U} p_{j,i} \log\left(1 + \frac{y_i}{\lambda_i} \right) \right| \leq \sum_{i \in U} |p_{j,i}|.$$

We thus have

$$\sum_{i \in U} \alpha_{j,i} r_{ki} \lambda_i + \sum_{i \in U} p_{j,i} \log(\lambda_i) < C + \sum_{i \in U} |\alpha_{j,i}| r_{ki} + \sum_{i \in U} |p_{j,i}| \leq C'$$

and hence, Eq. (8.7) holds.   ∎

Together Claim 8.13 and Claim 8.12 reformulate the logical condition of Lemma 8.10 concluding the proof of Lemma 8.11.   □

By testing the LC condition and the condition from Lemma 8.11 for each possible $X, \mathcal{Y}, k, U$, in turn using the (conditionally decidable) first-order theory of the reals with exponential function, we have:

Lemma 8.14. *Given a weighted automaton $\mathcal{W}$ and states s, s' such that $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ are plus-letter-bounded, it is decidable whether s is big-O s', subject to Schanuel's conjecture.*

8.4. **The letter-bounded case.** Here we consider the case where $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ are letter-bounded, $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ are subsets of $a_1^* \dots a_m^*$ for some $a_1, \dots, a_m \in \Sigma$, which is a relaxation of the preceding case. For the plus-letter-bounded case, we relied on a 1-1 correspondence between numeric vectors and words. This correspondence no longer holds in the letter-bounded case: for example, a^n matches $a^* b^* a^*$, but it could correspond to $(n, 0, 0)$, $(0, 0, n)$, as well as any $(n_1, 0, n_2)$ with $n_1 + n_2 = n$. Still, there is a reduction to the plus-letter-bounded case.

Lemma 8.15. *The big-O problem for $\mathcal{W}, s, s'$ with $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ letter-bounded reduces to the plus-letter-bounded case.*

Proof. Suppose the LC condition holds and $\mathcal{L}_s(\mathcal{W}) \subseteq \mathcal{L}_{s'}(\mathcal{W}) \subseteq a_1^* \cdots a_m^*$. Let I be the set of strictly increasing sequences $\vec{\imath} = i_1 \cdots i_k$ of integers between 1 and m. Given $\vec{\imath} \in I$, let $\mathcal{W}_{\vec{\imath}}$ be the weighted automaton obtained by intersecting $\mathcal{W}$ with a DFA for $a_{i_1}^+ \cdots a_{i_k}^+$ whose initial state is q. Note that s is big-O of s' (in $\mathcal{W}$) iff (s, q) is big-O of (s', q) in $\mathcal{W}_{\vec{\imath}}$ for all $\vec{\imath} \in I$, because $a_1^* \cdots a_m^* = \bigcup_{\vec{\imath} \in I} a_{i_1}^+ \cdots a_{i_k}^+$. Because the big-O problem for each $\mathcal{W}_{\vec{\imath}}$, (s, q), (s', q) falls into the plus-letter-bounded case, the results follows from Lemma 8.14.   □

8.5. **The bounded case.** Here we consider the case where $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ are bounded, which is a relaxation of letter-boundedness (see Definition 2.6): $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ are subsets of $w_1^* \dots w_m^*$ for some $w_1, \dots, w_m \in \Sigma^*$. We show a reduction to the letter-bounded case from Section 8.4.

Lemma 8.16. *The big-O problem for $\mathcal{W}, s, s'$ with $\mathcal{L}_s(\mathcal{W})$ and $\mathcal{L}_{s'}(\mathcal{W})$ bounded reduces to the letter-bounded case.*

Suppose $\mathcal{W}$ is bounded over $w_1^* \dots w_m^*$, our approach is to construct a new weighted automaton $\mathcal{W}'$ letter-bounded over a new alphabet $a_1^* \dots a_m^*$ with the following property. For *every* decomposition of a word w, as $w_1^{n_1} \dots w_m^{n_m}$, the weight of $a_1^{n_1} \dots a_m^{n_m}$ in $\mathcal{W}'$ is equal to the weight of w in $\mathcal{W}$.

To showcase the difference to the letter-bounded case, consider the language $(abab)^* a^* b^* (ab)^*$. Observe that, for example the word $(ab)^4$ can be decomposed in a number of ways: $(abab)^2 a^0 b^0 (ab)^0$, $(abab)^1 a^1 b^1 (ab)^1$, $(abab)^1 a^0 b^0 (ab)^2$, $(abab)^0 a^1 b^1 (ab)^3$ or $(abab)^0 a^0 b^0 (ab)^4$. One must be careful to consider all such decompositions.

Proof of Lemma 8.16. Let $\mathcal{W} = \langle Q, \Sigma, M, F \rangle$. Then we have $w_1, \ldots, w_m$ such that for all w with $f_s(w) > 0$, $w = w_1^{n_1} \ldots w_m^{n_m}$ for some $n_1, \ldots, n_m \in \mathbb{N}$. Let us assume $w_i = b_{i,1} b_{i,2}, \ldots, b_{i,|w_i|}$.

Given a word w, there may be multiple paths $\pi_1, \pi_2, \ldots$ from s to t respecting that word. Further there may be multiple decomposition vectors $\vec{n}_1, \vec{n}_2, \cdots \in \mathbb{N}^m$ such that $\vec{n}_i = (n_1, \ldots, n_m)$ and $w = w_1^{n_1} \ldots w_m^{n_m}$. Our goal will be to construct a weighted automaton $\mathcal{W}'$ with states $\hat{s}$ and $\hat{s}'$ letter-bounded over $a_1^* \ldots a_m^*$ such that, for every word w, the weight of $a_1^{n_1} \ldots a_m^{n_m}$ in $\mathcal{W}'$ from $\hat{s}$ (resp. $\hat{s}'$), for every valid decomposition vector $\vec{n} \in \mathbb{N}^m$ of w, will be the sum of the weights of all paths $\pi_1, \pi_2, \ldots$ respecting w in $\mathcal{W}$ from s (resp. s'). To compute $\mathcal{W}'$, we will define a transducer and apply it to our automaton $\mathcal{W}$.

A nondeterministic finite transducer is an NFA with transitions labelled by pairs from $\Sigma \times (\Sigma' \cup \{\varepsilon\})$, denoted by a/b for $a \in \Sigma$ and $b \in \Sigma' \cup \{\varepsilon\}$. In our construction, we only require edges of this form, i.e. we do not consider a definition with transitions labelled with ε in the first component (e.g., ε/a). Our transducer induces a translation $\mathcal{T}: \Sigma^* \to \Sigma'^*$.

Consider the set of regular expressions $w_{i_1}^+ \ldots w_{i_{m'}}^+$ each induced by a sequence $\vec{i} = (i_1, \ldots, i_{m'}) \in \mathbb{N}^{m'}$, $m' \leq m$, with $1 \leq i_1 < \cdots < i_{m'} \leq m$. Note that two sequences $(i_1', \ldots, i_{m'}'), (i_1'', \ldots, i_{m''}'')$ may yield the same expression $w_{i_1}^+ \ldots w_{i_m}^+$, in which case we need not consider more than one. The transducer $\mathcal{T}$ will be defined as follows.

For each $\vec{i} = (i_1, \ldots, i_{m'})$ described above, build the following automaton. For each i_j, construct the following section, which simply reads the word w_{i_j}:

$$f_j^{\vec{i}} \xrightarrow{b_{i_j,1}/\varepsilon} s_j^{\vec{i}} \xrightarrow{b_{i_j,2}/\varepsilon} \cdot \xrightarrow{b_{i_j,3}/\varepsilon} \cdot \ldots \cdot \xrightarrow{b_{i_j,|w_i|-1}/\varepsilon} e_j^{\vec{i}}.$$

Then, on the final character, nondeterministically restart or move to the next word, emitting a character representing the word:

$$e_j^{\vec{i}} \xrightarrow{b_{i_j,|w_{i_j}|}/a_{i_j}} f_j^{\vec{i}} \quad \text{and} \quad e_j^{\vec{i}} \xrightarrow{b_{i_j,|w_{i_j}|}/a_{i_j}} f_{j+1}^{\vec{i}}.$$

The transducer $\mathcal{T}$ is defined by the union of the above transitions over all $\vec{i}$. We also add a global start state q_0, from which we would like to move nondeterministically to $f_1^{\vec{i}}$ for each $\vec{i}$. To achieve this and avoid ε transitions, we duplicate the transitions $f_1^{\vec{i}} \xrightarrow{x} s_1^{\vec{i}}$ with $q_0 \xrightarrow{x} s_1^{\vec{i}}$. Observe that the valid output sequences are $(\varepsilon^* a_1)^* (\varepsilon^* a_2)^* \ldots (\varepsilon^* a_m)^*$.

Assume $\mathcal{W} = \langle Q, \Sigma, M, \{t\} \rangle$ and $\mathcal{T} = \langle Q', \Sigma \times (\Sigma' \cup \{\varepsilon\}), \to, q_0 \rangle$. Then construct the weighted automaton $\mathcal{T}(\mathcal{W}) = \langle Q \times Q', \Sigma', M^{\mathcal{T}}, \{t\} \times Q' \rangle$ using a product construction. The probability is associated in the following way $M^{\mathcal{T}}(a)((s, q), (s', q')) = p$ if there is a transition $q \xrightarrow{b/a} q'$ in $\mathcal{T}$ and $s \xrightarrow{p}{b} s'$ in $\mathcal{W}$. Note that, by this definition, there is a matrix $M^{\mathcal{T}}(\varepsilon)$; however, in every run of $\mathcal{T}(\mathcal{W})$ there can be a finite number of ε's in a row, at most $r = \max_{1 \leq i \leq m} |w_i| - 1$.

Now let $\mathcal{W}'$ be a copy of $\mathcal{T}(\mathcal{W})$ with ε removed: $M'(a_i) = (\sum_{x=0}^{r} M^{\mathcal{T}}(\varepsilon)^x) M^{\mathcal{T}}(a^i)$. Then $f_{\mathcal{W}}(w) = f_{\mathcal{W}'}(a_1^{n_1} \ldots a_m^{n_m})$ for all $n_1, \ldots, n_m$ such that $w = w_1^{n_1} \ldots w_m^{n_m}$. Hence, $\mathcal{W}'$ is a weighted automaton with letter-bounded languages from (s, q_0) and (s', q_0) such that (s, q_0) is big-O of (s', q_0) in $\mathcal{W}'$ if and only if s is big-O of s' in $\mathcal{W}$. $\square$

9. Analysis by Ambiguity

When a problem is shown to be undecidable, it is often only undecidable for relatively complex instances. We have already considered the problem when the language of the weighted automaton is restricted to bounded languages. It is also interesting to consider whether decision problem are decidable depending on the ambiguity of the instances [FRW17, DJL+21].

We say a weighted automaton $\mathcal{W}$ is *unambiguous from a state s* if every word has at most one accepting path in $\mathcal{N}_s(\mathcal{W})$. A weighted automaton $\mathcal{W}$ is *finitely ambiguous from a state s* if there exists a constant k such that every word has at most k accepting paths in $\mathcal{N}_s(\mathcal{W})$. A weighted automaton may also be polynomially ambiguous, if there exists a polynomial $p(\cdot)$ such that every word w has at most $p(|w|)$ accepting paths in $\mathcal{N}_s(\mathcal{W})$. If a weighted automaton is not polynomially ambiguous, then it is *exponentially ambiguous*.

Theorem 4.2 proves that the big-O problem is undecidable. However, the weighted automaton resulting from the reduction, for which the big-O problem cannot be decided, is exponentially ambiguous. This raises the question whether it is decidable for restricted ambiguities. We answer this positively, showing polynomial time decidability for unambiguous weighted automata, and decidability for finite ambiguous weighted automata subject to Schanuel's conjecture (Example 8.2 is finitely ambiguous).

We leave open whether the big-O problem is decidable for polynomially ambiguous weighted automata.

9.1. Unambiguous weighted automata.
In this section, we prove the polynomial-time solvability in the unambiguous case.

Theorem 9.1. *If a weighted automaton $\mathcal{W}$ is unambiguous from states s and s', the big-O problem is decidable in polynomial time.*

Proof. Let $\mathcal{W} = \langle Q, \Sigma, M, F \rangle$ be a weighted automaton. Suppose $s, s' \in Q$, t is a unique final state, and $\mathcal{W}$ is unambiguous from s, s'.

If $\mathcal{W}$ fails the LC condition (recall that it can be checked in polynomial time), we return no. Otherwise, let us construct a weighted automaton $\mathcal{W}'$ through a restricted product construction involving two copies of $\mathcal{W}$: for all $q_1, q_2, q_1', q_2' \in Q$, we add edges $(q_1, q_1') \xrightarrow[a]{p} (q_2, q_2')$ provided $M(a)(q_1, q_2) > 0$, $M(a)(q_1', q_2') > 0$ and $p = \frac{M(a)(q_1, q_2)}{M(a)(q_1', q_2')}$. Note that there exists a positively-weighted w-labelled path from (s, s') to (t, t) in $\mathcal{W}'$ iff $w \in \mathcal{L}_s(\mathcal{W}) \cap \mathcal{L}_{s'}(\mathcal{W})$. By the LC condition, this is equivalent to $w \in \mathcal{L}_s(\mathcal{W})$, and, to examine the big-O problem, it suffices to consider only such words.

By unambiguity of $\mathcal{W}$ from s and s', for any $w \in \mathcal{L}_s(\mathcal{W})$, there can be exactly one positively-weighted path from (s, s') to (t, t) in $\mathcal{W}'$. Consequently, the product of weights along this path is equal to $f_s(w)/f_{s'}(w)$. Hence, s is not big-O of s' (for $\mathcal{W}$) if and only there exists a positively-weighted path from (s, s') to (t, t) in $\mathcal{W}'$ that contains a cycle such that the product of the weights in that cycle is greater than 1.

Thus, to decide the big-O problem for s, s', it suffices to be able to detect such cycles. This can be done, for instance, by a modified version of the Bellman-Ford algorithm [CLR90] applied to the weighted directed graph consisting of positively-weighted edges of $\mathcal{W}'$. The algorithm is normally used to find negative cycles in the sense that the sum of weights is negative. To adapt it to our setting, we can apply the logarithm function to the weights.

However, to preserve rationality of weights and polynomial-time complexity, we cannot afford to do that explicitly. Instead, whenever $\log(x) < \log(y)$ would be tested, we test $x < y$ and, whenever $\log(x) + \log(y)$ would be performed, we compute xy instead. $\qquad\square$

Note the relevant behaviours are those on cycles—transitions which are taken at most once are of little significance to the big-O problem. Such transitions have at most a constant multiplicative effect on the ratio. This is the case whether or not the system is unambiguous.

9.2. **Finitely ambiguous weighted automata.** A weighted automaton is finitely ambiguous if there exists a constant k such that for every word w, there are at most k accepting paths in $\mathcal{N}_\mathcal{A}(\mathcal{W})$. Here show the following result for the Big-O problem on finitely ambiguous weighted automata, a problem left open in [CKMP20]:

Theorem 9.2. *Given two finitely ambiguous weighted automata, $\mathcal{A}, \mathcal{B}$, it is decidable, subject to Schanuel's conjecture, whether $\mathcal{A}$ is big-O of $\mathcal{B}$.*

Theorem 9.2 is proven by encoding the big-O problem into the first order theory of the reals with exponential function in Lemma 9.5. As a consequence, the Big-O problem is decidabile, conditional on Schanuel's conjecture (recall from Section 8.1 that Schanuel's conjecture entails decidability of $\mathrm{Th}(\mathbb{R}_{\exp})$).

Daviaud et al. [DJL+21] show, in Proposition 11 (and Proposition 15 therein), the following characterisation of attainable weights for pairs of finitely ambiguous weighted automata, which they use in the inclusion problem. In the following we adopt the notation $f_\mathcal{A}$ for the weight from the dedicated starting state of weighted automaton $\mathcal{A}$ and similarly for a weighted automaton $\mathcal{B}$.

Theorem 9.3 ([DJL+21]). *Given two finitely ambiguous weighted automata, $\mathcal{A}, \mathcal{B}$, one can compute a finite set of tuples Δ with elements $(p, q^1, \ldots, q^k, r, s^1, \ldots, s^\ell)$ where $p \in \mathbb{Q}_{>0}^k, r \in \mathbb{Q}_{>0}^\ell$, $q^i, s^i \in \mathbb{Q}_{>0}^m$ for some m, such that:*

- *for all $(p, q^1, \ldots, q^k, r, s^1, \ldots, s^\ell) \in \Delta$ and $n \in \mathbb{N}^m$ there exists a word $w \in \Sigma^*$ such that*

$$f_\mathcal{A}(w) = \sum_{i=1}^k p_i (q_1^i)^{n_1} \ldots, (q_m^i)^{n_m} \quad and \quad f_\mathcal{B}(w) = \sum_{i=1}^\ell r_i (s_1^i)^{n_1} \ldots, (s_m^i)^{n_m}; \qquad (9.1)$$

- *for all words $w \in \Sigma^*$ there exist $(p, q^1, \ldots, q^k, r, s^1, \ldots s^\ell) \in \Delta$ and $n \in \mathbb{N}^m$ such that Equation (9.1) holds.*

Since Δ is a finite set, if there is a sequence of words $(w_i)_i$ such that $\frac{f_\mathcal{A}(w_i)}{f_\mathcal{B}(w_i)} \to \infty$, then there is an infinite subsequence such that every w_i can be associated with a single choice of $(p, q^1, \ldots q^k, r, s^1, \ldots s^\ell) \in \Delta$. As such, we can characterise the big-O condition using the following logical characterisation:

Lemma 9.4. *$\mathcal{A}$ is not big-O of $\mathcal{B}$ if and only if there exists $(p, q^1, \ldots q^k, r, s^1, \ldots s^\ell) \in \Delta$, such that for all $C > 0$, there exists $n \in \mathbb{N}^m$ such that*

$$\sum_{i=1}^k p_i (q_1^i)^{n_1} \ldots (q_m^i)^{n_m} \geq C \sum_{i=1}^\ell r_i (s_1^i)^{n_1} \ldots (s_m^i)^{n_m}.$$

Since we can iterate through $(p, q^1, \ldots q^k, r, s^1, \ldots s^\ell) \in \Delta$, it remains to show that it is possible to check the logical characterisation. Like in Claim 8.13, we show this logical characterisation using naturals is equivalent to a statement that can be encoded in $\mathrm{Th}(\mathbb{R}_{\exp})$, the theory of the reals with exponential function.

Lemma 9.5. $\mathcal{A}$ *is not big-O of* $\mathcal{B}$ *if and only if there exists* $(p, q^1, \ldots q^k, r, s^1, \ldots s^\ell) \in \Delta$, *such that for all* $C > 0$, *there exists* $x \in \mathbb{R}_{\geq 0}^m$ *such that*

$$\sum_{i=1}^{k} p_i (q_1^i)^{x_1} \ldots (q_m^i)^{x_m} \geq C \sum_{i=1}^{\ell} r_i (s_1^i)^{x_1} \ldots (s_m^i)^{x_m}.$$

Proof. Clearly if the real formulation is unsatisfied, then the formulation with naturals is unsatisfied. It remains to show that if the real formulation is satisfied, then so too is the formulation with naturals. We assume the condition in Lemma 9.5 is satisfied for $(p, q^1, \ldots q^k, r, s^1, \ldots s^\ell) \in \Delta$ fixed and we show:

$$\forall C > 0, \ \exists n \in \mathbb{N}^m \text{ s.t. } \sum_{i=1}^{k} p_i (q_1^i)^{n_1} \ldots (q_m^i)^{n_m} \geq C \sum_{i=1}^{\ell} r_i (s_1^i)^{n_1} \ldots (s_m^i)^{n_m}.$$

Let C be given, we show the existence of a relevant vector n. Let us choose $C' = TC$, the exact value of $T \in \mathbb{R}_+$ will be chosen later, so that by assumption there exists $x \in \mathbb{R}_{\geq 0}^m$ such that

$$\sum_{i=1}^{k} p_i (q_1^i)^{x_1} \ldots (q_m^i)^{x_m} \geq C' \sum_{i=1}^{\ell} r_i (s_1^i)^{x_1} \ldots (s_m^i)^{x_m}. \tag{9.2}$$

We decompose x into its integer and fractional parts. Let n, y be such that $x = n + y$, $0 \leq y < 1$ and $n \in \mathbb{N}^m$.

Let $A = \max\limits_{1 \leq i \leq k} \prod\limits_{1 \leq j \leq m} \max\limits_{0 \leq z \leq 1} (q_j^i)^z$ and observe that

$$\sum_{i=1}^{k} p_i (q_1^i)^{x_1} \ldots (q_m^i)^{x_m} = \sum_{i=1}^{k} p_i (q_1^i)^{n_1} \ldots (q_m^i)^{n_m} (q_1^i)^{y_1} \ldots (q_m^i)^{y_m}$$

$$\leq A \sum_{i=1}^{k} p_i (q_1^i)^{n_1} \ldots (q_m^i)^{n_m}. \tag{9.3}$$

Similarly, let $B = \min\limits_{1 \leq i \leq \ell} \prod\limits_{1 \leq j \leq m} \min\limits_{0 \leq z \leq 1} (s_j^i)^z$, and observe that

$$\sum_{i=1}^{\ell} r_i (s_1^i)^{x_1} \ldots (s_m^i)^{x_m} = \sum_{i=1}^{\ell} r_i (s_1^i)^{n_1} \ldots (s_m^i)^{n_m} (s_1^i)^{y_1} \ldots (s_m^i)^{y_m}$$

$$\geq B \sum_{i=1}^{\ell} r_i (s_1^i)^{n_1} \ldots (s_m^i)^{n_m}. \tag{9.4}$$

Finally, letting $T = \frac{A}{B}$ we can conclude that

$$\sum_{i=1}^{k} p_i (q_1^i)^{n_1} \ldots (q_m^i)^{n_m} \geq \frac{1}{A} \sum_{i=1}^{k} p_i (q_1^i)^{x_1} \ldots (q_m^i)^{x_m} \qquad \text{by Eq. (9.3)}$$

$$\geq \frac{1}{A} C' \sum_{i=1}^{\ell} r_i (s_1^i)^{x_1} \ldots (s_m^i)^{x_m} \qquad \text{by Eq. (9.2)}$$

$$\geq \frac{C'}{A} B \sum_{i=1}^{\ell} r_i (s_1^i)^{n_1} \ldots (s_m^i)^{n_m} \qquad \text{by Eq. (9.4)}$$

$$\geq C \sum_{i=1}^{\ell} r_i (s_1^i)^{n_1} \ldots (s_m^i)^{n_m}, \qquad \text{as } C' = C\frac{A}{B}$$

as required. $\qquad\qquad \square$

10. Conclusion

Despite undecidability results, we have identified several decidable cases of the big-O problem. However, for bounded languages, the result depends on a conjecture from number theory, leaving open the exact borderline between decidability and undecidability.

Natural directions for future work include the analogous problem for infinite words, further analysis on ambiguity (e.g., is the big-O problem decidable for polynomially-ambiguous weighted automata?), and the extension to negative edge weights.

Acknowledgement

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`.

[ACOW18] Shaull Almagor, Dmitry Chistikov, Joël Ouaknine, and James Worrell. O-minimal invariants for linear loops. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 114:1–114:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPIcs.ICALP.2018.114`.

[BCV18] Suguman Bansal, Swarat Chaudhuri, and Moshe Y. Vardi. Comparator automata in quantitative verification. In *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018*, volume 10803 of *Lecture Notes in Computer Science*, pages 420–437. Springer, 2018. `doi:10.1007/978-3-319-89366-2_23`.

[BGS09] Andrew Badr, Viliam Geffert, and Ian Shipman. Hyper-minimizing minimized deterministic finite state automata. *RAIRO Theor. Informatics Appl.*, 43(1):69–94, 2009. `doi:10.1051/ita:2007061`.

[BPR05] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. Betti number bounds, applications and algorithms. *Current trends in combinatorial and computational geometry: papers from the special program at MSRI*, 52:87–97, 2005.

[BPR06]   Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and computation in mathematics*. Springer, 2nd edition, 2006. `doi: 10.1007/3-540-33099-2`.

[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`.

[CGPX14]   Konstantinos Chatzikokolakis, Daniel Gebler, Catuscia Palamidessi, and Lili Xu. Generalized Bisimulation Metrics. In *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014*, volume 8704 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2014. `doi:10.1007/978-3-662-44584-6_4`.

[Chr86]   Marek Chrobak. Finite automata and unary languages. *Theor. Comput. Sci.*, 47(3):149–158, 1986. `doi:10.1016/0304-3975(86)90142-8`.

[CK14]   Taolue Chen and Stefan Kiefer. On the total variation distance of labelled Markov chains. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS 2014*, pages 33:1–33:10. ACM, 2014. `doi:10.1145/2603088.2603099`.

[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. `doi:10.4230/LIPIcs.CONCUR.2020.41`.

[CKV14]   Rohit Chadha, Dileep Kini, and Mahesh Viswanathan. Decidable problems for unary PFAs. In *Quantitative Evaluation of Systems - 11th International Conference, QEST 2014*, volume 8657 of *Lecture Notes in Computer Science*, pages 329–344. Springer, 2014. `doi:10.1007/978-3-319-10696-0_26`.

[CLR90]   T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[CMP19]   Dmitry Chistikov, Andrzej S. Murawski, and David Purser. Asymmetric distances for approximate differential privacy. In *30th International Conference on Concurrency Theory, CONCUR 2019*, volume 140 of *LIPIcs*, pages 10:1–10:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.CONCUR.2019.10`.

[Coh13]   Henri Cohen. *A course in computational algebraic number theory*, volume 138. Springer Science & Business Media, 2013.

[Col15]   Thomas Colcombet. Unambiguity in automata theory. In *Descriptional Complexity of Formal Systems - 17th International Workshop, DCFS 2015*, volume 9118 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2015. `doi:10.1007/978-3-319-19225-3_1`.

[COW16]   Ventsislav Chonev, Joël Ouaknine, and James Worrell. On the Skolem problem for continuous linear dynamical systems. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*, volume 55 of *LIPIcs*, pages 100:1–100:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPIcs.ICALP.2016.100`.

[dB81]   N. G. de Bruijn. *Asymptotic methods in analysis*. Dover, 1981.

[DJKV17]   Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. A storm is coming: A modern probabilistic model checker. In *Computer Aided Verification - 29th International Conference, CAV 2017*, volume 10427 of *Lecture Notes in Computer Science*, pages 592–600. Springer, 2017. `doi:10.1007/978-3-319-63390-9_31`.

[DJL⁺21]   Laure Daviaud, Marcin Jurdziński, Ranko Lazić, Filip Mazowiecki, Guillermo A. Pérez, and James Worrell. When are emptiness and containment decidable for probabilistic automata? *Journal of Computer and System Sciences*, 119:78–96, 2021. `doi:10.1016/j.jcss.2021.01.006`.

[DMNS06]   Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. `doi:10.1007/11681878_14`.

[ESY84]   Shimon Even, Alan L. Selman, and Yacov Yacobi. The complexity of promise problems with applications to public-key cryptography. *Inf. Control.*, 61(2):159–173, 1984. `doi:10.1016/S0019-9958(84)80056-X`.

[FGO12]   Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012*, pages 295–304. IEEE Computer Society, 2012. `doi:10.1109/LICS.2012.40`.

[FGR15]   Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. Quantitative languages defined by functional automata. *Log. Methods Comput. Sci.*, 11(3), 2015. `doi:10.2168/LMCS-11(3:14)2015`.

[Fij17]   Nathanaël Fijalkow. Undecidability results for probabilistic automata. *SIGLOG News*, 4(4):10–17, 2017. `doi:10.1145/3157831.3157833`.

[FRW17]   Nathanaël Fijalkow, Cristian Riveros, and James Worrell. Probabilistic automata of bounded ambiguity. In *28th International Conference on Concurrency Theory, CONCUR 2017*, volume 85 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. `doi:10.4230/LIPIcs.CONCUR.2017.19`.

[FS80]   Shmuel Friedland and Hans Schneider. The growth of powers of a nonnegative matrix. *SIAM J. Matrix Analysis Applications*, 1(2):185–200, 1980. `doi:10.1137/0601022`.

[FT10]   Olivier Finkel and Stevo Todorčević. The isomorphism relation between tree-automatic structures. *Central European Journal of Mathematics*, 8:299–313, 2010.

[Gin66]   Seymour Ginsburg. *The Mathematical Theory of Context-Free Languages*. McGraw-Hill, 1966.

[GKRS10]   Pawel Gawrychowski, Dalia Krieger, Narad Rampersad, and Jeffrey Shallit. Finding the growth rate of a regular or context-free language in polynomial time. *Int. J. Found. Comput. Sci.*, 21(4):597–618, 2010. `doi:10.1142/S0129054110007441`.

[GO10]   Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010*, volume 6199 of *Lecture Notes in Computer Science*, pages 527–538. Springer, 2010. `doi:10.1007/978-3-642-14162-1_44`.

[GS64]   Seymour Ginsburg and Edwin H Spanier. Bounded ALGOL-like languages. *Transactions of the American Mathematical Society*, 113(2):333–368, 1964.

[HLXL18]   Cheng-Chao Huang, Jing-Cao Li, Ming Xu, and Zhi-Bin Li. Positive root isolation for poly-powers by exclusion and differentiation. *Journal of Symbolic Computation*, 85:148–169, 2018. 41th International Symposium on Symbolic and Algebraic Computation (ISSAC'16). `doi:10.1016/j.jsc.2017.07.007`.

[HT03]   Thanh Minh Hoang and Thomas Thierauf. The complexity of the characteristic and the minimal polynomial. *Theor. Comput. Sci.*, 295:205–222, 2003. `doi:10.1016/S0304-3975(02)00404-8`.

[IMNN14]   Egor Ianovski, Russell G. Miller, Keng Meng Ng, and André Nies. Complexity of equivalence relations and preorders from computability theory. *J. Symb. Log.*, 79(3):859–881, 2014. `doi:10.1017/jsl.2013.33`.

[Kie18]   Stefan Kiefer. On computing the total variation distance of hidden Markov models. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 130:1–130:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPIcs.ICALP.2018.130`.

[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. *Logical Methods in Computer Science*, 9(1), 2013. `doi:10.2168/LMCS-9(1:8)2013`.

[KNP11]   Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Computer Aided Verification - 23rd International Conference, CAV 2011*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011. `doi:10.1007/978-3-642-22110-1_47`.

[Kro94]   Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, 4:405–425, 1994. `doi:10.1142/S0218196794000063`.

[Lan66]   Serge Lang. *Introduction to transcendental numbers*. Addison-Wesley Pub. Co., 1966.

[Mal11]   Andreas Maletti. Notes on hyper-minimization. In *Automata and Formal Languages, 13th International Conference, AFL 2011*, pages 34–49, 2011.

[Mar02]   Andrew Martinez. Efficient computation of regular expressions from unary NFAs. In *Fourth International Workshop on Descriptional Complexity of Formal Systems - DCFS 2002*, volume Report No. 586, pages 174–187. Department of Computer Science, The University of Western Ontario, Canada, 2002.

[Mes19a] David Mestel. Quantifying information flow in interactive systems. In *32nd IEEE Computer Security Foundations Symposium, CSF 2019*, pages 414–427. IEEE, 2019. `doi:10.1109/CSF.2019.00035`.

[Mes19b] David Mestel. Widths of Regular and Context-Free Languages. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019*, volume 150 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.FSTTCS.2019.49`.

[Mig82]   M. Mignotte. *Some Useful Bounds*, pages 259–263. Springer Vienna, Vienna, 1982. `doi:10.1007/978-3-7091-3406-1_16`.

[MS72]   Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th Annual Symposium on Switching and Automata Theory, 1972*, pages 125–129. IEEE Computer Society, 1972.

[MSS20]   Rupak Majumdar, Mahmoud Salamati, and Sadegh Soudjani. On decidability of time-bounded reachability in CTMDPs. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 133:1–133:19, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ICALP.2020.133`.

[MW96]   Angus Macintyre and Alex J. Wilkie. On the decidability of the real exponential field. In Piergiorgio Odifreddi, editor, *Kreiseliana. About and Around Georg Kreisel*, pages 441–467. A K Peters, 1996.

[OW14]   Joël Ouaknine and James Worrell. Positivity problems for low-order linear recurrence sequences. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 366–379. SIAM, 2014. `doi:10.1137/1.9781611973402.27`.

[Pan96]   Victor Y Pan. Optimal and nearly optimal algorithms for approximating polynomial zeros. *Computers & Mathematics with Applications*, 31(12):97–138, 1996.

[Par66]   Rohit J Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966. `doi:10.1145/321356.321364`.

[Paz14]   Azaria Paz. *Introduction to probabilistic automata*. Academic Press, 2014.

[Poo03]   Bjorn Poonen. Hilbert's tenth problem over rings of number-theoretic interest. *Note from the lecture at the Arizona Winter School on "Number Theory and Logic"*, 2003. URL: `https://math.mit.edu/~poonen/papers/aws2003.pdf`.

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

[Sch86]   Hans Schneider. The influence of the marked reduced graph of a nonnegative matrix on the Jordan form and on related properties: A survey. *Linear Algebra and its Applications*, 84:161–189, 1986.

[Sén93]   Géraud Sénizergues. An effective version of Stallings' theorem in the case of context-free groups. In *Automata, Languages and Programming, 20nd International Colloquium, ICALP 1993*, volume 700 of *Lecture Notes in Computer Science*, pages 478–495. Springer, 1993. `doi:10.1007/3-540-56939-1_96`.

[Ser13]   Bruno Sericola. *Markov chains: theory and applications*. John Wiley & Sons, 2013.

[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`.

[Smi08]   Adam D. Smith. Efficient, Differentially Private Point Estimators. *CoRR*, abs/0809.4794, 2008. URL: `http://arxiv.org/abs/0809.4794`.

[To09]   Anthony Widjaja To. Unary finite automata vs. arithmetic progressions. *Inf. Process. Lett.*, 109(17):1010–1014, 2009. `doi:10.1016/j.ipl.2009.06.005`.

[Tze92]   Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM J. Comput.*, 21(2):216–227, 1992. `doi:10.1137/0221017`.

[Tze96]   Wen-Guey Tzeng. On path equivalence of nondeterministic finite automata. *Inf. Process. Lett.*, 58(1):43–46, 1996. `doi:10.1016/0020-0190(96)00039-7`.

[Wal00]   Michel Waldschmidt. *Diophantine Approximation on Linear Algebraic Groups*, volume 326 of *Grundlehren der mathematischen Wissenschaften (A Series of Comprehensive Studies in Mathematics)*. Springer, Berlin, Heidelberg, 2000.