# When is Containment Decidable for Probabilistic Automata?


**Laure Daviaud**
University of Warwick, Coventry, UK
L.Daviaud@warwick.ac.uk

**Marcin Jurdziński**
University of Warwick, Coventry, UK
Marcin.Jurdzinski@warwick.ac.uk

**Ranko Lazić**[1]
University of Warwick, Coventry, UK
R.S.Lazic@warwick.ac.uk

**Filip Mazowiecki**[2]
Université de Bordeaux, Bordeaux, France
filip.mazowiecki@u-bordeaux.fr

**Guillermo A. Pérez**[3]
Université libre de Bruxelles, Brussels, Belgium
gperezme@ulb.ac.be
https://orcid.org/0000-0002-1200-4952

**James Worrell**[4]
University of Oxford, Oxford, UK
James.Worrell@cs.ox.ac.uk
https://orcid.org/0000-0001-8151-2443



## Abstract

The containment problem for quantitative automata is the natural quantitative generalisation of the classical language inclusion problem for Boolean automata. We study it for probabilistic automata, where it is known to be undecidable in general. We restrict our study to the class of probabilistic automata with bounded ambiguity. There, we show decidability (subject to Schanuel's conjecture) when one of the automata is assumed to be unambiguous while the other one is allowed to be finitely ambiguous. Furthermore, we show that this is close to the most general decidable fragment of this problem by proving that it is already undecidable if one of the automata is allowed to be linearly ambiguous.



**2012 ACM Subject Classification** Theory of computation → Quantitative automata, Theory of computation → Probabilistic computation

**Keywords and phrases** Probabilistic automata, Containment, Emptiness, Ambiguity

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2018.121



[1] R. Lazić has been supported by a Leverhulme Trust Research Fellowship RF-2017-579.
[2] F. Mazowiecki has been supported by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux (ANR-10-IDEX-03-02).
[3] G. A. Pérez has been supported by an F.R.S.-FNRS Aspirant fellowship and an FWA postdoc fellowship.
[4] J. Worrell has been supported by the EPSRC Fellowship EP/N008197/1.




45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 121; pp. 121:1–121:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**Related Version** A full version of the paper is available at https://arxiv.org/abs/1804.09077.

**Funding** L. Daviaud, M. Jurdziński, and R. Lazić have been supported by the EPSRC grant EP/P020992/1.

**Acknowledgements** We thank Shaull Almagor for some helpful remarks.

## 1 Introduction

Probabilistic automata (PA) are a quantitative extension of classical Boolean automata that were first introduced by Rabin [20]. Non-deterministic choices are replaced by probabilities: each transition carries a rational number which gives its probability to be chosen amongst all the other transitions going out of the same state and labelled by the same letter. Then, instead of simply accepting or rejecting a word, such an automaton measures the probability of it being accepted.

PA can be seen as (blind) partially observable Markov decision processes [19]. The latter have numerous applications in the field of artificial intelligence [22, 11]. Further applications for PA include, amongst others, verification of probabilistic systems [23, 14, 5], reasoning about inexact hardware [18], quantum complexity theory [25], uncertainty in runtime modelling [9], as well as text and speech processing [17]. PA are very expressive, as witnessed by the mentioned applications, most natural verification-related decision problems for them are consequently undecidable. However, equivalence and minimisation do admit efficient algorithms [13].

Due to the aforementioned negative results, many sub-classes of probabilistic automata have been studied. These include hierarchical [7] and leaktight [2] automata; and more recently, bounded-ambiguity automata [8] (see [6] for a survey).

In this paper, we continue the study of the class of PA with bounded ambiguity. We focus on the containment problem: *given two automata $\mathcal{A}$ and $\mathcal{B}$, determine whether for all words $w$, the probability of it being accepted by $\mathcal{A}$ is at most the probability of it being accepted by $\mathcal{B}$.* The problem is known to be undecidable even for the subclass of automata with polynomial ambiguity, more specifically, already for automata with quadratic ambiguity [8].

**Contributions.** In this paper, we refine the undecidability result by extending it to the class of linearly ambiguous automata.

▶ **Theorem 1.** *The containment problem is undecidable for the class of linearly ambiguous probabilistic automata.*

The proof we provide gives in fact two stronger results. Firstly, the containment problem for linearly ambiguous PA is already undecidable if one of the two input automata is unambiguous. Secondly, and perhaps more importantly, the better-known emptiness problem (*given a probabilistic automaton, does there exist a word accepted with probability at least* $1/2$*?*) is also undecidable for the class of linearly ambiguous PA. This strictly refines the previous best known result [8].

This negative result motivates us to turn our attention to the class of finitely ambiguous PA. For this class, we prove that the containment problem is decidable, provided that one of the two input automata is unambiguous (and conditional on Schanuel's conjecture).

▶ **Theorem 2.** *If Schanuel's conjecture holds then the containment problem is decidable for the class of finitely ambiguous probabilistic automata, provided that at least one of the input automata is unambiguous.*

The intermediate problem, i.e., when both input PA are finitely ambiguous, remains open.

**Organisation of the paper.** In Section 2, we give the formal definition of probabilistic automata, the notion of ambiguity, and the problems under consideration. We also recall classical results that will be useful in the paper. In Section 3, we explain how to translate the containment problem into a problem about the existence of integral exponents for certain exponential inequalities. Using this formalism, we prove that the containment problem for $\mathcal{A}$ and $\mathcal{B}$, as stated above, is decidable if $\mathcal{A}$ is finitely ambiguous and $\mathcal{B}$ is unambiguous. In Section 4, we tackle the more challenging direction and prove that the containment problem is also decidable if $\mathcal{A}$ is unambiguous and $\mathcal{B}$ is finitely ambiguous. Finally, in Section 5, we prove that the containment problem is undecidable provided that one of the automata is linearly ambiguous.

## 2 Preliminaries

In this section, we define probabilistic automata and recall some classical results.

**Notation.** We use boldface lower-case letters, e.g., $\boldsymbol{a}, \boldsymbol{b}, \ldots$, to denote vectors and upper-case letters, e.g., $M, N, \ldots$, for matrices. For a vector $\boldsymbol{a}$, we write $a_i$ for its $i$-th component, and $\boldsymbol{a}^\top$ for its transpose.

### 2.1 Probabilistic automata and ambiguity

For a finite set $S$, we say that a function $f : S \to \mathbb{Q}_{\geq 0}$ is a *distribution over* $S$ if $\sum_{s\in S} f(s) \leq 1$. We write $\mathcal{D}(S)$ for the set of all distributions over $S$. We also say that a vector $\boldsymbol{d} = (d_1, d_2, \ldots, d_n) \in \mathbb{Q}^n_{\geq 0}$ of non-negative rationals is a distribution if $\sum_{i=1}^n d_i \leq 1$.

A *probabilistic automaton (PA)* $\mathcal{A}$ is a tuple $(\Sigma, Q, \delta, \iota, F)$, where:

- $\Sigma$ is the finite alphabet,
- $Q$ is the finite set of states,
- $\delta : Q \times \Sigma \to \mathcal{D}(Q)$ is the (probabilistic) transition function,
- $\iota \in \mathcal{D}(Q)$ is the initial distribution, and
- $F \subseteq Q$ is the set of final states.

We write $\delta(q, a, p)$ instead of $\delta(q, a)(p)$ for the *probability of moving from $q$ to $p$ reading $a$*. Consider the word $w = a_1 \ldots a_n \in \Sigma^*$. A *run $\rho$ of $\mathcal{A}$ over $w = a_1 \ldots a_n$* is a sequence of transitions $(q_0, a_1, q_1), (q_1, a_2, q_2), \ldots, (q_{n-1}, a_n, q_n)$ where $\delta(q_{i-1}, a_i, q_i) > 0$ for all $1 \leq i \leq n$. It is an *accepting run* if $\iota(q_0) > 0$ and $q_n \in F$. The *probability of the run* $\rho$ is $\Pr_{\mathcal{A}}(\rho) \stackrel{\text{def}}{=} \iota(q_0) \cdot \prod_{i=1}^n \delta(q_{i-1}, a_i, q_i)$.

The automaton $\mathcal{A}$ realizes a function $[\![\mathcal{A}]\!]$ mapping words over the alphabet $\Sigma$ to values in $[0, 1]$. Formally, for all $w \in \Sigma^*$, we set: $[\![\mathcal{A}]\!](w) \stackrel{\text{def}}{=} \sum_{\rho\in \mathrm{Acc}_{\mathcal{A}}(w)} \Pr_{\mathcal{A}}(\rho)$ where $\mathrm{Acc}_{\mathcal{A}}(w)$ is the set of all accepting runs of $\mathcal{A}$ over $w$.

**Ambiguity.** The notion of ambiguity depends only on the structure of the underlying automaton (i.e., whether a probability is null or not, but not on its actual value). An automaton $\mathcal{A}$ is said to be *unambiguous* (resp. *$k$-ambiguous*) if for all words $w$, there is at most one accepting run (resp. $k$ accepting runs) over $w$ in $\mathcal{A}$. If an automaton is $k$-ambiguous for some $k$, then it is said to be *finitely ambiguous*. If there exists a polynomial $P$, such that

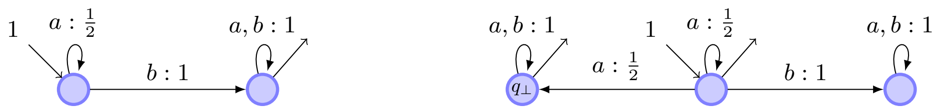


**Figure 1** Two PA over the alphabet $\Sigma = \{a, b\}$ are depicted. On the left hand side, automaton $\mathcal{A}$ induces the function $a^n b\Sigma^* \mapsto \frac{1}{2^n}$ and $a^* \mapsto 0$. On the right hand side, the automaton $\overline{\mathcal{A}}$ induces the function $a^n b\Sigma^* \mapsto 1 - \frac{1}{2^n}$ and $a^* \mapsto 1$. Observe that $\mathcal{A}$ is unambiguous and $\overline{\mathcal{A}}$ is linearly ambiguous.

for every word $w$, the number of accepting runs of $\mathcal{A}$ on $w$ is bounded by $P(|w|)$ (where $|w|$ is the length of $w$), then $\mathcal{A}$ is said to be *polynomially ambiguous*, and *linearly ambiguous* whenever the degree of $P$ is at most 1.

It is well-known that if an automaton is not finitely ambiguous then it is at least linearly ambiguous (see, for example, the criterion in [24, Section 3]). The same paper shows that if an automaton is finitely ambiguous then it is $k$-ambiguous for $k$ bounded exponentially in the number of states of that automaton.

We give two examples of PA and discuss their ambiguity in Figure 1. As usual, they are depicted as graphs. The initial distribution is denoted by ingoing arrows associated with their probability (when there is no such arrow, the initial probability is 0) and the final states are denoted by outgoing arrows.

## 2.2 Decision problems

In this work, we are interested in comparing the functions computed by PA. We write $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ if "$\mathcal{A}$ is contained in $\mathcal{B}$", that is if $[\![\mathcal{A}]\!](w) \leq [\![\mathcal{B}]\!](w)$ for all $w \in \Sigma^*$; and we write $[\![\mathcal{A}]\!] < \frac{1}{2}$ if $[\![\mathcal{A}]\!](w) < \frac{1}{2}$ for all $w \in \Sigma^*$. We are interested in the following decision problems for PA.

- *Containment problem:* Given probabilistic automata $\mathcal{A}$ and $\mathcal{B}$, does $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ hold?
- *Emptiness problem:* Given a probabilistic automaton $\mathcal{A}$, does $[\![\mathcal{A}]\!] < \frac{1}{2}$ hold?

We will argue that the containment and emptiness problems are both undecidable when considered for the class of linearly ambiguous automata (Section 5). The emptiness problem is known to be decidable for the class of finitely ambiguous automata [8]. We tackle here the more difficult containment problem (Sections 3 and 4).

## 2.3 Classical results

**Weighted-sum automaton.** For PA $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ over the same alphabet, and for a discrete distribution $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$, the *weighted sum* (of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ with weights $\boldsymbol{d}$) is defined to be the disjoint union of the $n$ automata with the initial distribution $\iota(q) \stackrel{\text{def}}{=} d_i \cdot \iota_i(q)$ if $q$ is a state of $\mathcal{A}_i$, where $\iota_i$ is the initial distribution of $\mathcal{A}_i$. Note that if $\mathcal{B}$ is the weighted sum of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ with weights $\boldsymbol{d}$ then it is also a probabilistic automaton and $[\![\mathcal{B}]\!] = \sum_{i=1}^{n} d_i \cdot [\![\mathcal{A}_i]\!]$.

**Complement automaton.** For a PA $\mathcal{A}$, we define its *complement automaton* $\overline{\mathcal{A}}$ in the following way. First, define the PA $\mathcal{A}'$ by modifying $\mathcal{A}$ as follows:

- add a new sink state $q_\bot$;
- obtain the transition function $\delta'$ from $\delta$ by adding transitions:
  - $\delta'(q_\bot, a, q_\bot) = 1$ for all $a \in \Sigma$,
  - $\delta'(q, a, q_\bot) = 1 - \sum_{r \in Q} \delta(q, a, r)$ for all $(q, a) \in Q \times \Sigma$;
- obtain the initial distribution $\iota'$ from $\iota$ by adding $\iota'(q_\bot) = 1 - \sum_{q \in Q} \iota(q)$.

Observe that $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$, that $\sum_{r\in Q} \delta'(q,a,r) = 1$ for all $(q,a) \in Q \times \Sigma$, and that $\sum_{q\in Q} \iota'(q) = 1$. We obtain $\overline{\mathcal{A}}$ from $\mathcal{A}'$ by swapping its final and non-final states. As expected, it is the case that $[\![\overline{\mathcal{A}}]\!] = 1 - [\![\mathcal{A}]\!]$.

▶ Remark (Preserving ambiguity). The ambiguity of a weighted-sum automaton is the sum of the ambiguities of the individual automata, and the ambiguity of a complement automaton may be larger than the ambiguity of the original one (see Figure 1).

## 3 Decidability of the case finitely ambiguous vs. unambiguous

Our aim is to decide whether $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$. We first give a translation of the problem into a problem about the existence of integral exponents for certain exponential inequalities.

**Notation.** In the rest of the paper, we write $\exp(x)$ to denote the exponential function $x \mapsto e^x$, and $\log(y)$ for the natural logarithm function $y \mapsto \log_e(y)$. For a real number $x$ and a positive real number $y$, we write $y^x$ for $\exp(x\log(y))$.

### 3.1 Translating the containment problem into exponential inequalities

We are going to translate the negation of the containment problem: Given two finitely ambiguous PA $\mathcal{A}$ and $\mathcal{B}$, does there exist a word $w$, such that $[\![\mathcal{A}]\!](w) > [\![\mathcal{B}]\!](w)$? Consider two positive integers $k$ and $n$, and vectors $\boldsymbol{p} \in \mathbb{Q}^k_{>0}$ and $\boldsymbol{q}_1, \dots, \boldsymbol{q}_k \in \mathbb{Q}^n_{>0}$. We denote by $S(\boldsymbol{p}, \boldsymbol{q}_1, \dots, \boldsymbol{q}_k) : \mathbb{N}^n \to \mathbb{R}$ the function associating a vector $\boldsymbol{x} \in \mathbb{N}^n$ to $\sum_{i=1}^k p_i q_{i,1}^{x_1} \dots q_{i,n}^{x_n}$, where $q_{i,j}$ is the $j$-th component of vector $\boldsymbol{q}_i$.

▶ **Proposition 3.** *Given a $k$-ambiguous automaton $\mathcal{A}$ and an $\ell$-ambiguous automaton $\mathcal{B}$, one can compute a positive integer $n$ and a finite set $\Delta$ of tuples $(\boldsymbol{p}, \boldsymbol{q}_1, \dots, \boldsymbol{q}_{k'}, \boldsymbol{r}, \boldsymbol{s}_1, \dots, \boldsymbol{s}_{\ell'})$ of vectors $\boldsymbol{p} \in \mathbb{Q}^{k'}_{>0}, \boldsymbol{r} \in \mathbb{Q}^{\ell'}_{>0}$, for some $k' \leq k$ and $\ell' \leq \ell$; and $\boldsymbol{q}_i \in \mathbb{Q}^n_{>0}, \boldsymbol{s}_j \in \mathbb{Q}^n_{>0}$, for all $i$ and $j$; such that the following two conditions are equivalent:*
- *there exists $w \in \Sigma^*$ such that $[\![\mathcal{A}]\!](w) > [\![\mathcal{B}]\!](w)$,*
- *there exist $(\boldsymbol{p}, \boldsymbol{q}_1, \dots, \boldsymbol{q}_{k'}, \boldsymbol{r}, \boldsymbol{s}_1, \dots, \boldsymbol{s}_{\ell'}) \in \Delta$ and $\boldsymbol{x} \in \mathbb{N}^n$ such that*

$$S(\boldsymbol{p}, \boldsymbol{q}_1, \dots, \boldsymbol{q}_{k'})(\boldsymbol{x}) > S(\boldsymbol{r}, \boldsymbol{s}_1, \dots, \boldsymbol{s}_{\ell'})(\boldsymbol{x}).$$

It thus follows that to prove Theorem 2, it suffices to show decidability of the second item of Proposition 3 for a given element of $\Delta$ in the cases where either $k$ or $\ell$ are equal to 1. To prove Proposition 3, the idea is to decompose every run into a short path and simple cycles using the well-known simple-cycle decomposition (see, e.g., [21]). We do this simultaneously for all runs in $\mathcal{A}$ and $\mathcal{B}$. The vectors $\boldsymbol{p}$ and $\boldsymbol{r}$ then correspond to probabilities of simple paths in every run; $\boldsymbol{q}_i$ (resp. $\boldsymbol{s}_j$), to all simple cycles in the $i$-th run in $\mathcal{A}$ (resp. $j$-th run in $\mathcal{B}$). Finally, $\boldsymbol{x}$ specifies how many times each simple cycle occurs in the decomposition.

▶ **Example 4.** Consider the following instance of the problem, where $k = n = 2$, $\ell = 1$, and $p$ is a fixed rational number $0 \leq p \leq 1$: Do there exist $x, y \in \mathbb{N}$ such that $p \cdot \left(\frac{1}{12}\right)^x \cdot \left(\frac{1}{2}\right)^y + (1-p) \cdot \left(\frac{1}{3}\right)^x \cdot \left(\frac{1}{18}\right)^y < \left(\frac{1}{6}\right)^x \cdot \left(\frac{1}{6}\right)^y$. This can be rewritten as

$$p \cdot \left(\frac{1}{2}\right)^x \cdot 3^y + (1-p) \cdot 2^x \cdot \left(\frac{1}{3}\right)^y < 1$$

or equivalently, using the exponential function, as follows

$$\exp(\log(p) - x\log(2) + y\log(3)) + \exp(\log(1-p) + x\log(2) - y\log(3)) < 1.$$

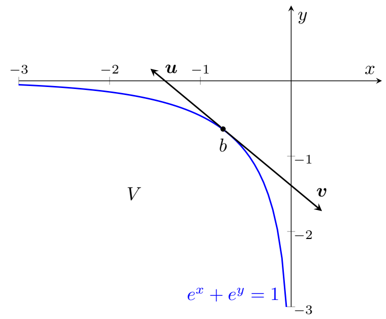


**Figure 2** The set $V$ is bounded by the plot $e^x + e^y = 1$ and the point $b$ is on that plot.

Consider the set $V = \{(x, y) \in \mathbb{R}^2 \mid e^x + e^y < 1\}$ and denote by $b$ the point $(\log(p), \log(1-p))$. Let $\boldsymbol{u} = (-\log(2), \log(2))$ and $\boldsymbol{v} = (\log(3), -\log(3))$ be two vectors. See Figure 2 for a geometric representation. The question is now: do there exist $x, y \in \mathbb{N}$ such that $b + x\boldsymbol{u} + y\boldsymbol{v} \in V$. We will show that the answer is yes if and only if $p \neq \frac{1}{2}$.

Let $C = \{(x, -x) \mid x \in \mathbb{R}\}$. For $p = \frac{1}{2}$, the affine line $C + p$ is tangent to the blue plot and so, whatever the values of $x$ and $y$, $b + x\boldsymbol{u} + y\boldsymbol{v}$ cannot be in $V$. For $p \neq \frac{1}{2}$, there is a value $\delta$ such that the whole interval strictly between $b$ and $b + (\delta, -\delta)$ is in $V$. Since $\log(2)$ and $\log(3)$ are rationally independent, the set $D = \{x\boldsymbol{u} + y\boldsymbol{v} \mid x, y \in \mathbb{N}\}$ is a dense subset of $C$, so in particular, there is a point of $D + b$ in the interval between $b$ and $b + (\delta, -\delta)$ and thus there exist $x, y \in \mathbb{N}$ such that $b + x\boldsymbol{u} + y\boldsymbol{v} \in V$.

## 3.2 Decidability

We prove here the decidability of the containment problem when $\mathcal{A}$ is finitely ambiguous and $\mathcal{B}$ is unambiguous. The converse situation is tackled in Section 4.

▶ **Proposition 5.** *Determining whether $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ is decidable when $\mathcal{A}$ is finitely ambiguous and $\mathcal{B}$ is unambiguous.*

**Proof.** Let $\mathcal{A}$ be $k$-ambiguous. Proposition 3 shows that it is sufficient to decide, given an integer $n$ and positive rational numbers $p, q_j, r_i, s_{i,j}$ for $i \in \{1, \ldots, k\}$, $j \in \{1, \ldots, n\}$, whether there exists $x_1, \ldots, x_n \in \mathbb{N}$ such that

$$\sum_{i=1}^{k} p_i q_{i,1}^{x_1} \cdots q_{i,n}^{x_n} > r s_1^{x_1} \cdots s_n^{x_n}. \tag{1}$$

We consider two cases. First, assume that there exist $i$ and $j$ such that $q_{i,j} > s_j$. Then in that case, for a large enough $m \in \mathbb{N}$ condition (1) will be satisfied for $(x_1, \ldots, x_j, \ldots, x_n) = (0, \ldots, m, \ldots, 0)$. Otherwise, assume that $\max\{q_{i,j} \mid 1 \leq i \leq k\} \leq s_j$ for all $1 \leq j \leq n$. In this case, if there exists a valuation of the $x_i$ satisfying (1) then $(x_1, \ldots, x_n) = (0, \ldots, 0)$ also satisfies it. It is then sufficient to test condition (1) for $x_1 = \cdots = x_n = 0$ to conclude. ◀

# 4 Decidability of the case unambiguous vs. finitely ambiguous

In this section we will show the more challenging part of Theorem 2, i.e., that the containment problem is decidable for $\mathcal{A}$ unambiguous and $\mathcal{B}$ finitely ambiguous. Our proof is conditional on the first-order theory of the reals with the exponential function being decidable. In [15], the authors show that this is the case if a conjecture due to Schanuel and regarding transcendental number theory is true.

▶ **Theorem 6.** *Determining whether* $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ *is decidable when* $\mathcal{A}$ *is unambiguous and* $\mathcal{B}$ *is finitely ambiguous, assuming Schanuel's conjecture is true.*

## 4.1 Integer programming problem with exponentiation

Given two positive integers $n$ and $\ell$, we define $\mathcal{F}_{n,\ell}$ to be the set of all the functions $f : \mathbb{R}^n \to \mathbb{R}$ such that there exist $\boldsymbol{r} \in \mathbb{Q}^{\ell}_{>0}$ and $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_\ell \in \mathbb{Q}^n_{>0}$ such that $f(\boldsymbol{x}) = \sum_{i=1}^{\ell} r_i s_{i,1}^{x_1} \ldots s_{i,n}^{x_n}$. Observe that this is just a lifting of the $S(\cdot)$ function, defined in the previous section, to real-valued parameters. Consider the following integer programming problem with exponentiation.

▶ **Problem 7** (IP+EXP).
***Input:*** *Three positive integers* $n$, $\ell$ *and* $m$, *a function* $f \in \mathcal{F}_{n,\ell}$, *a matrix* $M \in \mathbb{Z}^{m\times n}$, *and a vector* $\boldsymbol{c} \in \mathbb{Z}^m$.
***Question:*** *Does there exist* $\boldsymbol{x} \in \mathbb{Z}^n$ *such that* $f(\boldsymbol{x}) < 1$ *and* $M\boldsymbol{x} < \boldsymbol{c}$*?*

In the sequel, we will show that the above problem is decidable.

▶ **Theorem 8.** *The IP+EXP problem is decidable, assuming Schanuel's conjecture is true.*

Theorem 6 is a direct corollary of Theorem 8.

**Proof of Theorem 6.** Proposition 3 shows that, in order to prove Theorem 6, it is sufficient to decide, given an integer $n$ and positive rational numbers $p, r_i, q_j, s_{i,j}$ for $i \in \{1, \ldots, \ell\}$, $j \in \{1, \ldots, n\}$, whether there exist $x_1, \ldots, x_n \in \mathbb{N}$ such that $pq_1^{x_1} \cdots q_n^{x_n} > \sum_{i=1}^{\ell} r_i s_{i,1}^{x_1} \cdots s_{i,n}^{x_n}$ or equivalently, whether there exist $x_1, \ldots, x_n \in \mathbb{N}$ such that:

$$\sum_{i=1}^{\ell} r_i p^{-1}(s_{i,1}q_1^{-1})^{x_1} \cdots (s_{i,n}q_n^{-1})^{x_n} < 1. \tag{2}$$

Define $f : \mathbb{R}^n \to \mathbb{R}$ such that $f(\boldsymbol{x}) = \sum_{i=1}^{\ell} r_i p^{-1}(s_{i,1}q_1^{-1})^{x_1} \cdots (s_{i,n}q_n^{-1})^{x_n}$. Then, inequality (2) becomes $f(\boldsymbol{x}) < 1$. We can now apply Theorem 8 with $m$ set to be $n$; $M$, to be $-Id$, where $Id$ is the identity matrix; and $\boldsymbol{c}$ to be the null vector. ◀

Since the IP+EXP problem is semi-decidable (indeed, we can enumerate the vectors $\boldsymbol{x}$ in $\mathbb{Z}^n$ to find one satisfying the conditions), it will suffice to give a semi-decision procedure to determine whether the inequalities $f(\boldsymbol{x}) < 1 \wedge M\boldsymbol{x} < \boldsymbol{c}$ have no integer solution. We give now such a procedure.

## 4.2 Semi-decision procedure for the complement of IP+EXP

Consider as input for the IP+EXP problem three positive integers $n$, $\ell$, $m$, a function $f \in \mathcal{F}_{n,\ell}$, a matrix $M \in \mathbb{Z}^{m\times n}$, and a vector $\boldsymbol{c} \in \mathbb{Z}^m$. Denote by $X$ the set of *real* solutions of the problem, i.e., the set of vectors $X = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) < 1 \wedge M\boldsymbol{x} < \boldsymbol{c}\}$.

**Proc($n$, $\ell$, $m$, $f$, $M$, $\boldsymbol{c}$):**

1. Search for a non-zero vector $\boldsymbol{d} \in \mathbb{Z}^n$ and $a, b \in \mathbb{Z}$ such that $\{\boldsymbol{d}^\top \boldsymbol{x} \mid \boldsymbol{x} \in X\} \subseteq [a, b]$. Set $i = a$.
2. If $i > b$, then stop and return `YES`. Otherwise, let $Y_i$ be the set of vectors $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying $d_1x_1 + \cdots + d_nx_n = i$. If $Y_i$ is empty, then increment $i$ and start again from step 2. Otherwise:
   a. Compute $N \in \mathbb{Z}^{n\times(n-1)}$ and $\boldsymbol{h} \in \mathbb{Z}^n$ such that $Y_i = \{N\boldsymbol{y} + \boldsymbol{h} \mid \boldsymbol{y} \in \mathbb{Z}^{n-1}\}$.

**b.** If $n-1=0$ and $f(\boldsymbol{h})<1 \wedge M\boldsymbol{h}<\boldsymbol{c}$ then return NO, otherwise increment $i$ and start again from step 2.

**c.** If $n-1>0$ then recursively call $\mathbf{Proc}(n-1,\ \ell,\ m,\ f',\ M',\ \boldsymbol{c}')$, where $f' \in \mathcal{F}_{n-1,\ell}$ is defined as $f'(\boldsymbol{y}) = f(N\boldsymbol{y}+\boldsymbol{h})$; $M' \in \mathbb{Z}^{m\times(n-1)}$, as $M' = MN$; and $\boldsymbol{c}' \in \mathbb{Z}^m$, as $\boldsymbol{c}-M\boldsymbol{h}$. If the procedure stops and returns YES then increment $i$ and start again from step 2. If the procedure stops and returns NO then return NO.

▶ **Lemma 9.** *The above semi-decision procedure stops and outputs* YES *if and only if there is no integer valuation of $\boldsymbol{x}$ that satisfies the constraints, i.e. $X \cap \mathbb{Z}^n$ is empty.*

First, notice that the only step which might not terminate in a call to our procedure is step 1. Indeed, once $\boldsymbol{d}$, $a$, and $b$ are fixed, there are only finitely many integers $i \in [a,b]$ that have to be considered in step 2. For each of them, one can compute in a standard way (see, e.g., [3]) the set of integer solutions of the equation $d_1x_1+\cdots+d_nx_n = i$ and the corresponding $N$ and $\boldsymbol{h}$ as in the procedure.

Moreover, for each integer vector $\boldsymbol{d} \in \mathbb{Z}^n$ and $a,b \in \mathbb{Z}$, the inclusion $\{\boldsymbol{d}^\top\boldsymbol{x} \mid \boldsymbol{x} \in X\} \subseteq [a,b]$ that needs to be checked in step 1 can be formulated as a decision problem in the first-order logic over the structure $(\mathbb{R},+,\times,\exp)$. Since this structure has a decidable first-order theory subject to Schanuel's conjecture [15], the inclusion can be decided for each fixed $\boldsymbol{d}$, $a$, and $b$.

To prove Lemma 9 we use the two following lemmata. The first one is the most technical contribution of the paper and is proved in Section 4.3. It ensures termination of step 1 in the procedure when there is no integer solution.

▶ **Lemma 10.** *If the set $X$ contains no integer point then there must exist a non-zero integer vector $\boldsymbol{d} \in \mathbb{Z}^n$ and $a,b \in \mathbb{Z}$ such that $\{\boldsymbol{d}^\top\boldsymbol{x} \mid \boldsymbol{x} \in X\} \subseteq [a,b]$.*

▶ **Lemma 11.** *Given a non-zero vector $\boldsymbol{d} \in \mathbb{Z}^n$ and an integer $i$, there exists $\boldsymbol{x} \in \mathbb{Z}^n$ such that $f(\boldsymbol{x})<1 \wedge M\boldsymbol{x}<\boldsymbol{c} \wedge \boldsymbol{d}^\top\boldsymbol{x} = i$ if and only if there exists $\boldsymbol{y} \in \mathbb{Z}^{n-1}$ such that $f'(\boldsymbol{y})<1 \wedge M'\boldsymbol{y}<\boldsymbol{c}'$ where $f'$, $M'$ and $\boldsymbol{c}'$ are as defined in the procedure.*

We can now prove Lemma 9.

**First direction: when the procedure returns YES.** Suppose first that the semi-decision procedure stops and outputs YES. Then there exist a non-zero vector $\boldsymbol{d} \in \mathbb{Z}^n$ and $a,b \in \mathbb{Z}$ such that $\{\boldsymbol{d}^\top\boldsymbol{x} \mid \boldsymbol{x} \in X\} \subseteq [a,b]$ as in step 1, and for all integers $i \in [a,b]$, one of the following situations occurs:

1. $Y_i$ is empty,
2. $n-1=0$, $Y_i = \{\boldsymbol{h}\}$ as defined in step 2.$a$ but $\boldsymbol{h}$ is not an integer solution of the problem,
3. $n-1>0$ and the recursive call stops and outputs YES.

By definition of $\boldsymbol{d}$, in order to prove that there is no integer solution of the problem, we need to show that in all those cases, and for all $i \in [a,b]$, $Y_i \cap X = \emptyset$. It is clear for items 1 and 2 and we use Lemma 11 and an induction for item 3.

**Second direction: when $\boldsymbol{X} \cap \mathbb{Z}^{\boldsymbol{n}} = \emptyset$.** If there is no integer solution then by Lemma 10, there must exist a non-zero vector $\boldsymbol{d} \in \mathbb{Z}^n$ and $a,b \in \mathbb{Z}$ such that $\{\boldsymbol{d}^\top\boldsymbol{x} \mid \boldsymbol{x} \in X\} \subseteq [a,b]$ as in step 1. Moreover, for any of those choices, if for an integer $i \in [a,b]$, the set $Y_i$ of vectors $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying $d_1x_1+\cdots+d_nx_n = i$ is non-empty, then,

1. if $n=1$, then $\boldsymbol{h}$ as defined in step 2.$a$, is not a solution of the problem (by hypothesis) and thus the procedure stops and returns YES,
2. if $n>1$, we use Lemma 11 and, by induction, the recursive call must return YES.

## 4.3 Proof of Lemma 10

Fix three positive integers $n$, $\ell$, $m$, a function $f \in \mathcal{F}_{n,\ell}$, a matrix $M \in \mathbb{Z}^{m\times n}$, and a vector $\boldsymbol{c} \in \mathbb{Z}^m$. Recall that we denote by $X$ the set of vectors

$$X = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) < 1 \wedge M\boldsymbol{x} < \boldsymbol{c}\}.$$

We want to prove that if the set $X$ contains no integer point then there must exist a non-zero integer vector $\boldsymbol{d} \in \mathbb{Z}^n$ and $a, b \in \mathbb{Z}$ such that $\{\boldsymbol{d}^\top \boldsymbol{x} \mid \boldsymbol{x} \in X\} \subseteq [a, b]$.

We will use the following corollary of Kronecker's theorem on simultaneous Diophantine approximation. It generalises the fact that any line in the plane with irrational slope passes arbitrarily close to integer points in the plane.

▶ **Proposition 12.** *[12, Corollary 2.8]. Let $\boldsymbol{u}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_s$ be vectors in $\mathbb{R}^n$. Suppose that for all $\boldsymbol{d} \in \mathbb{Z}^n$ we have $\boldsymbol{d}^\top \boldsymbol{u} = 0$ whenever $\boldsymbol{d}^\top \boldsymbol{u}_1 = \cdots = \boldsymbol{d}^\top \boldsymbol{u}_s = 0$. Then for all $\varepsilon > 0$ there exist real numbers $\lambda_1, \ldots, \lambda_s \geq 0$ and a vector $\boldsymbol{v} \in \mathbb{Z}^n$ such that $\|\boldsymbol{u} + \sum_{i=1}^{s} \lambda_i \boldsymbol{u}_i - \boldsymbol{v}\|_\infty \leq \varepsilon$.*

By definition, there exist vectors $\boldsymbol{r} \in \mathbb{Q}^\ell_{>0}$ and $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_\ell \in \mathbb{Q}^n_{>0}$ such that $f(\boldsymbol{x}) = \sum_{i=1}^{\ell} r_i s_{i,1}^{x_1} \ldots s_{i,n}^{x_n}$. Let $\boldsymbol{a} \in \mathbb{R}^\ell$ and $\boldsymbol{b}_i \in \mathbb{R}^n$ be defined by $a_i = \log(r_i)$ and $\boldsymbol{b}_i = (\log(s_{i,1}), \ldots, \log(s_{i,n}))$. We can then rewrite $f(\boldsymbol{x})$ as follows

$$f(\boldsymbol{x}) = \exp(\boldsymbol{b}_1^\top \boldsymbol{x} + a_1) + \cdots + \exp(\boldsymbol{b}_\ell^\top \boldsymbol{x} + a_\ell).$$

Let us now consider the cone

$$C = \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \boldsymbol{b}_1^\top \boldsymbol{x} \leq 0 \wedge \cdots \wedge \boldsymbol{b}_\ell^\top \boldsymbol{x} \leq 0 \wedge M\boldsymbol{x} \leq 0\right\}. \tag{3}$$

It is easy to see that $X + C \subseteq X$.

▶ **Lemma 13.** *Suppose that $X$ is non-empty and that no non-zero integer vector in $\mathbb{Z}^n$ is orthogonal to $C$. Then $X \cap \mathbb{Z}^n$ is non-empty.*

**Proof.** Let $\boldsymbol{u} \in X$. Since $X$ is open, there exists $\varepsilon > 0$ such that the open ball $B_\varepsilon(\boldsymbol{u})$ is contained in $X$. We therefore have that $B_\varepsilon(\boldsymbol{u}) + C \subseteq X$.

We will apply Proposition 12 to show that $B_\varepsilon(\boldsymbol{u}) + C$ contains an integer point and hence that $X$ contains an integer point. To this end, let vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s \in C$ be such that $\text{span}\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s\} = \text{span}(C)$. Then no non-zero vector in $\mathbb{Z}^n$ is orthogonal to $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s$. By Proposition 12, there exist real numbers $\lambda_1, \ldots, \lambda_s \geq 0$ and an integer vector $\boldsymbol{v} \in \mathbb{Z}^n$ such that $\|\boldsymbol{u} + \sum_{i=1}^{s} \lambda_i \boldsymbol{u}_i - \boldsymbol{v}\|_\infty \leq \varepsilon$. Thus, $\boldsymbol{v} \in B_\varepsilon(\boldsymbol{u}) + C \subseteq X$. ◀

The contrapositive of the above result states that if $X$ contains no integer point, then there must exist an integer vector that is orthogonal to $C$. For the desired result, it remains for us to prove the boundedness claim.

▶ **Lemma 14.** *Suppose that $\boldsymbol{d} \in \mathbb{Z}^n$ is orthogonal to the cone $C$. Then $\{\boldsymbol{d}^\top \boldsymbol{u} \mid \boldsymbol{u} \in X\}$ is bounded.*

**Proof.** Define the "enveloping polygon" of $X$ to be

$$\widehat{X} = \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \boldsymbol{b}_1^\top \boldsymbol{x} + \boldsymbol{a}_1 \leq 0 \wedge \cdots \wedge \boldsymbol{b}_\ell^\top \boldsymbol{x} + \boldsymbol{a}_\ell \leq 0 \wedge M\boldsymbol{x} \leq \boldsymbol{c}\right\}.$$

Clearly it holds that $X \subseteq \widehat{X}$. Moreover, by the Minkowski-Weyl decomposition theorem we can write $\widehat{X}$ as a sum $\widehat{X} = B + C$ for $B$ a bounded polygon and $C$ the cone defined in (3). Since $\boldsymbol{d}$ is orthogonal to $C$ by assumption, it follows that $\{\boldsymbol{d}^\top \boldsymbol{u} \mid \boldsymbol{u} \in \widehat{X}\} = \{\boldsymbol{d}^\top \boldsymbol{u} \mid \boldsymbol{u} \in B\}$ is bounded and hence $\{\boldsymbol{d}^\top \boldsymbol{u} \mid \boldsymbol{u} \in X\}$ is bounded. The result immediately follows. ◀

We can now complete the proof of Lemma 10.

**Proof of Lemma 10.** By Lemma 13 there exists a non-zero integer vector $\boldsymbol{d} \in \mathbb{Z}^n$ such that $\boldsymbol{d}$ is orthogonal to the cone $C$ defined in (3). Then by Lemma 14 we obtain that $\{\boldsymbol{d}^\top \boldsymbol{u} \mid \boldsymbol{u} \in X\}$ is contained in a bounded interval. ◀

## 5 Undecidability for linearly ambiguous automata

In this section we prove Theorem 1. That is, we argue that the containment problem is undecidable for the class of linearly ambiguous PA. For most of this section we deal with the following variant of the problem: does there exist a word $w$ such that $[\![\mathcal{A}]\!](w) \leq [\![\mathcal{B}]\!](w)$. Towards the end, we shortly explain how our reduction proves undecidability for any variant of this problem (that is, with inequality $<$). Moreover, we derive from the proof that all variants of the emptiness problem are undecidable for a given linearly ambiguous automaton $\mathcal{A}$, i.e. whether there exists $w$ such that $[\![\mathcal{A}]\!](w) \geq \frac{1}{2}$ (or resp. $>$, $\leq$, $<$).

The proof is done by a reduction from the halting problem for two-counter machines. The reduction resembles the one used to prove undecidability of the comparison problem for another quantitative extension of Boolean automata: max-plus automata [4, 1]. We give here an outline of the proof and the main ideas used in the reduction.

**Two-counter machines.** A *two-counter machine* (or a *Minsky machine*) is a deterministic finite-state machine, with an initial state, a final state, and two counters, each of which can be incremented or decremented (if its value is not 0). For each counter $i \in \{1, 2\}$, there are two types of transitions: $T_i^+ \subseteq Q^2$ such that a transition $(p, q) \in T_i^+$ moves from state $p$ to state $q$ and increments the $i$-th counter; and $T_i^- \subseteq Q^3$ such that a transition $(p, q, r) \in T_i^-$ moves from state $p$ to state $q$ if the $i$-th counter has value 0, and moves from state $p$ to state $r$ decrementing the $i$-th counter otherwise. The set of all transitions is denoted $T = T_1^+ \cup T_2^+ \cup T_1^- \cup T_2^-$.

The machine *halts* if there is a (unique) computation from the initial state to the final state. It is well-known that deciding whether a given machine halts is undecidable [16]. We use this fact to prove undecidability of the containment problem.

▶ **Proposition 15.** *Given a two-counter machine, one can construct two linearly ambiguous probabilistic automata $\mathcal{A}$ and $\mathcal{B}$, such that the machine halts if and only if there exists a word $w$ such that $[\![\mathcal{A}]\!](w) \leq [\![\mathcal{B}]\!](w)$.*

**Outline of the reduction.** The general idea is to encode executions of the two-counter machine into words over the alphabet $\Sigma = \{a, b\} \cup T$. A block $a^m$ (resp. $b^m$) encodes the fact that the value of the first (resp. second) counter is $m$. For example, given $t \in T_1^+$ and $t' \in T_2^-$, a word $a^n b^m t a^{n+1} b^m t' a^{n+1} b^{m'}$, encodes an execution starting with value $n$ in the first counter and $m$ in the second counter. Transition $t$ then increases the value of the first counter to $n + 1$ without changing the value of the second one. The configuration is thus encoded by the infix $a^{n+1} b^m$. Next, transition $t'$ is taken, and either $m' = m = 0$ or $m' = m - 1$. Moreover, if $t = (p, q)$ and $t' = (r, s, u)$ then $q = r$ (i.e., the states between transitions have to match).

**Simulating executions faithfully.** Our reduction has the following property: $[\![\mathcal{A}]\!](w) = [\![\mathcal{B}]\!](w)$ for the unique word $w$ encoding the halting execution (if it exists), and $[\![\mathcal{A}]\!](w) > [\![\mathcal{B}]\!](w)$ for all other words. This suffices to prove Proposition 15.

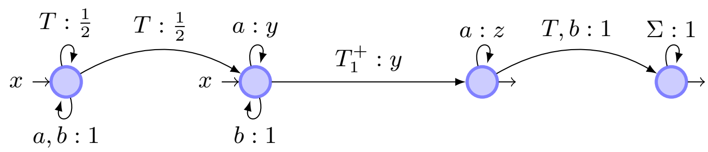


**Figure 3** Gadget automaton $\mathcal{C}(x, y, z)$ used to check if the first counter is incremented properly.

There are several conditions that the constructed automata $\mathcal{A}$ and $\mathcal{B}$ need to check to ensure the correctness of the encoding: whether the states match; whether the counter values encoded by blocks of $a$ and $b$ are increased and decreased properly, etc. It is easy to define an automaton $\mathcal{A}_0$ that will allow us to focus only on *proper words* of the form

$$w = a^{n_1} b^{m_1} t_1 a^{n_2} b^{m_2} t_2 a^{n_3} b^{m_3} t_3 \cdots a^{n_k} b^{m_k} t_k. \tag{4}$$

That is, $\mathcal{A}_0$ is such that $[\![\mathcal{A}_0]\!](w) = 0$ if the word $w$ correctly alternates between blocks of $a$, $b$, and letters from $T$; it correctly implements the zero tests within decrement transitions; it observes the initial and final states and checks that the states of adjacent transitions match. Observe that the latter are all regular conditions. If $w$ does not satisfy all these constraints, then $[\![\mathcal{A}_0]\!](w) = 1$.

For other (non-regular) conditions we construct pairs of automata $\mathcal{A}_i$ and $\mathcal{B}_i$, such that $[\![\mathcal{A}_i]\!](w) = [\![\mathcal{B}_i]\!](w)$ if a proper word $w$ does not violate the condition and $[\![\mathcal{A}_i]\!](w) > [\![\mathcal{B}_i]\!](w)$ for all other proper words. In the end, the automata $\mathcal{A}$ and $\mathcal{B}$ are obtained as weighted sums of the automata $\mathcal{A}_i$ and $\mathcal{B}_i$.

**Simulating increments.** We present the construction for one of the components of $\mathcal{A}$ and $\mathcal{B}$. The remaining ones are obtained using similar gadgets. The automata $\mathcal{A}_1$ and $\mathcal{B}_1$ check that a proper word encodes an execution where the first counter is always correctly incremented after reading transitions from $T_1^+$. Consider the automaton $\mathcal{C}(x, y, z)$ in Figure 3. It is parameterised by three probability variables $x, y, z > 0$. The parameter $x$ is the probability used by the initial distribution, and parameters $y$ and $z$ are used by some transitions.

Consider a word $w$ as in (4). Notice that the only non-deterministic transitions in $\mathcal{C}(x, y, z)$ are the ones going out from the the leftmost state upon reading letters from $T$. It follows that $\mathcal{C}(x, y, z)$ is linearly ambiguous. In fact, for every position in $w$ labelled by an element $t$ from $T_1^+$ there is a unique accepting run that first reaches a final state upon reading $t$. By construction, we have

$$[\![\mathcal{C}(x, y, z)]\!](w) = \sum_{t_i \in T_1^+} x \left(\frac{1}{2}\right)^{i-1} y^{n_i+1} z^{n_{i+1}}. \tag{5}$$

Let $x = \frac{1}{2}$, $y = 1$ and $z = \frac{1}{4}$. We define $\mathcal{B}_1$ as $\mathcal{C}(x, x, x)$ and $\mathcal{A}_1$ as a weighted sum of $\mathcal{C}(x, y, z)$ and $\mathcal{C}(x, z, y)$ with weights $(\frac{1}{2}, \frac{1}{2})$. Since $\mathcal{C}(\cdot, \cdot, \cdot)$ is linearly ambiguous the obtained automata are also linearly ambiguous. We prove that $[\![\mathcal{A}_1]\!](w) = [\![\mathcal{B}_1]\!](w)$ only if $n_i + 1 = n_{i+1}$ for all $i$ such that $t_i \in T_1^+$ and $[\![\mathcal{A}_1]\!](w) > [\![\mathcal{B}_1]\!](w)$ otherwise.

By (5) it suffices to show that for every $i$ it holds that

$$\left(\frac{1}{2}\right)^{n_i+1+n_{i+1}} \leq \frac{1}{2}\left(\left(\frac{1}{4}\right)^{n_{i+1}} + \left(\frac{1}{4}\right)^{n_i+1}\right)$$

and that the equality holds only if $n_i + 1 = n_{i+1}$, which is the case.

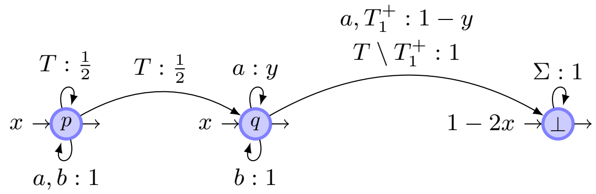


**Figure 4** Complement automaton of $\mathcal{C}(x, y, z)$ after trimming.

**Changes for version with strict inequalities.** Observe that if PA $\mathcal{A}$ and $\mathcal{B}$ output different probabilities on a word $w$ then the probabilities must differ by at least $x^{|w|+1}$ for some value $x$ depending on the probabilities used in $\mathcal{A}$ and $\mathcal{B}$ (see, e.g., [10]). By a weighted sum construction, we can define a PA $\mathcal{B}'$ associating a word $w$ to the value $[\![\mathcal{B}]\!](w) + x^{|w|+1}$. Then there is a word $w$ such that $[\![\mathcal{A}]\!](w) \leq [\![\mathcal{B}]\!](w)$ if and only if there is a word $w$ such that $[\![\mathcal{A}]\!](w) < [\![\mathcal{B}']\!](w)$.

**Changes for the emptiness problem.** Let us consider the containment problem $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$. For all words $w$, we have that $[\![\mathcal{A}]\!](w) \leq [\![\mathcal{B}]\!](w)$ holds if and only if $\frac{1}{2}[\![\mathcal{A}]\!](w) + \frac{1}{2}(1 - [\![\mathcal{B}]\!](w)) \leq \frac{1}{2}$. Thus, using the weighted sum and complement operations we can get a single probabilistic automaton that outputs $\frac{1}{2}[\![\mathcal{A}]\!](w) + \frac{1}{2}(1 - [\![\mathcal{B}]\!](w))$. Complementing an automaton may, in general, increase the ambiguity (see Figure 1). However, for the automata we have constructed, this is not the case. For example, it is easy to check that the complement automaton $\overline{\mathcal{C}(x, y, z)}$ depicted in Figure 4 is—just like $\mathcal{C}(x, y, z)$—also linearly ambiguous.

## 6 Conclusion

In this work we have shown that the containment problem for probabilistic automata is decidable if one of the automata is finitely ambiguous and the other one is unambiguous. Interestingly, for one of the two cases, our proposed algorithm uses a satisfiability oracle for a theory whose decidability is equivalent to a weak form of Schanuel's conjecture. We have complemented our decidability results with a proof of undecidability for the case when the given automata are linearly ambiguous.

Decidability of the containment problem when both automata are allowed to be finitely ambiguous remains open. One way to tackle it is to study generalizations of the IP+EXP problem introduced in Section 4. This problem asks whether there exists $\boldsymbol{x} \in \mathbb{N}^n$ such that $f(\boldsymbol{x}) < 1$ and $M\boldsymbol{x} < \boldsymbol{c}$ for a given function $f$ defined using exponentiations, a given matrix $M$, and vector $\boldsymbol{c}$. A natural way to extend the latter would be to ask that $f(\boldsymbol{x}) < g(\boldsymbol{x})$, where $g$ is obtained in a similar way as $f$. The main obstacle, when trying to generalize our decidability proof for that problem, is that we lack a replacement for the cone $C$ needed in order to obtain a result similar to Lemma 14 using the Minkowski-Weyl decomposition.

## References

1 Shaull Almagor, Udi Boker, and Orna Kupferman. What's decidable about weighted automata? In Tevfik Bultan and Pao-Ann Hsiung, editors, *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings*, volume 6996 of *Lecture Notes in Computer Science*, pages 482–491. Springer, 2011. doi:10.1007/978-3-642-24372-1_37.

2 Rohit Chadha, A. Prasad Sistla, Mahesh Viswanathan, and Yue Ben. Decidable and expressive classes of probabilistic automata. In Andrew M. Pitts, editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 200–214. Springer, 2015. doi:10.1007/978-3-662-46678-0_13.

3 Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate texts in mathematics*. Springer, 1993. URL: http://www.worldcat.org/oclc/27810276.

4 Thomas Colcombet. On distance automata and regular cost function. Presented at the Dagstuhl seminar "Advances and Applications of Automata on Words and Trees", 2010.

5 Lu Feng, Tingting Han, Marta Z. Kwiatkowska, and David Parker. Learning-based compositional verification for synchronous probabilistic systems. In Tevfik Bultan and Pao-Ann Hsiung, editors, *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011*, volume 6996 of *Lecture Notes in Computer Science*, pages 511–521. Springer, 2011. doi:10.1007/978-3-642-24372-1_40.

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

7 Nathanaël Fijalkow, Hugo Gimbert, Edon Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Logical Methods in Computer Science*, 11(2), 2015. doi:10.2168/LMCS-11(2:12)2015.

8 Nathanaël Fijalkow, Cristian Riveros, and James Worrell. Probabilistic automata of bounded ambiguity. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory, CONCUR 2017, September 5-8, 2017, Berlin, Germany*, volume 85 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.CONCUR.2017.19.

9 Holger Giese, Nelly Bencomo, Liliana Pasquale, Andres J. Ramirez, Paola Inverardi, Sebastian Wätzoldt, and Siobhán Clarke. Living with uncertainty in the age of runtime models. In Nelly Bencomo, Robert B. France, Betty H. C. Cheng, and Uwe Assmann, editors, *Models@run.time - Foundations, Applications, and Roadmaps [Dagstuhl Seminar 11481, November 27 - December 2, 2011]*, volume 8378 of *Lecture Notes in Computer Science*, pages 47–100. Springer, 2014. doi:10.1007/978-3-319-08915-7_3.

10 Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, volume 6199 of *Lecture Notes in Computer Science*, pages 527–538. Springer, 2010. doi:10.1007/978-3-642-14162-1_44.

11 Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996. doi:10.1613/jair.301.

12 Leonid Khachiyan and Lorant Porkolab. Computing integral points in convex semi-algebraic sets. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 162–171. IEEE Computer Society, 1997. doi:10.1109/SFCS.1997.646105.

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.

14 Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. Assume-guarantee verification for probabilistic systems. In Javier Esparza and Rupak Majumdar,

editors, *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 23–37. Springer, 2010. doi:10.1007/978-3-642-12002-2_3.

**15** 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. AK Peters, 1996.

**16** Marvin Lee Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

**17** Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002. doi:10.1006/csla.2001.0184.

**18** Krishna V. Palem and Lingamneni Avinash. Ten years of building broken chips: The physics and engineering of inexact computing. *ACM Transactions on Embedded Computing Systems*, 12(2s):87:1–87:23, 2013. doi:10.1145/2465787.2465789.

**19** Martin L. Puterman. *Markov Decision Processes*. Wiley-Interscience, 2005.

**20** Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963. doi:10.1016/S0019-9958(63)90290-0.

**21** Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Compututer Science*, 6:223–231, 1978. doi:10.1016/0304-3975(78)90036-1.

**22** Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010. URL: http://vig.pearsoned.com/store/product/1,1207,store-12521_isbn-0136042597,00.html.

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

**24** Andreas Weber and Helmut Seidl. On the degree of ambiguity of finite automata. *Theoretical Computer Science*, 88(2):325–349, 1991. doi:10.1016/0304-3975(91)90381-B.

**25** Abuzer Yakaryilmaz and A. C. Cem Say. Unbounded-error quantum computation with small space bounds. *Information and Computation*, 209(6):873–892, 2011. doi:10.1016/j.ic.2011.01.008.