DECIDING THE VALUE 1 PROBLEM
FOR PROBABILISTIC LEAKTIGHT AUTOMATA

NATHANAËL FIJALKOW, HUGO GIMBERT, EDON KELMENDI, AND YOUSSOUF OUALHADJ

LIAFA, Université Denis Diderot - Paris 7, France, and University of Warsaw, Poland.
e-mail address: nath@liafa.univ-paris-diderot.fr

LaBRI, CNRS, Bordeaux, France.
e-mail address: hugo.gimbert@labri.fr

LaBRI and Université de Bordeaux, France.
e-mail address: edon.kelmendi@labri.fr

Université Paris-Est, LACL, France.
e-mail address: youssouf.oualhadj@lacl.fr

ABSTRACT. The value 1 problem is a decision problem for probabilistic automata over finite words: given a probabilistic automaton, are there words accepted with probability arbitrarily close to 1? This problem was proved undecidable recently; to overcome this, several classes of probabilistic automata of different nature were proposed, for which the value 1 problem has been shown decidable. In this paper, we introduce yet another class of probabilistic automata, called *leaktight automata*, which strictly subsumes all classes of probabilistic automata whose value 1 problem is known to be decidable.

We prove that for leaktight automata, the value 1 problem is decidable (in fact, PSPACE-complete) by constructing a saturation algorithm based on the computation of a monoid abstracting the behaviours of the automaton. We rely on algebraic techniques developed by Simon to prove that this abstraction is complete. Furthermore, we adapt this saturation algorithm to decide whether an automaton is leaktight.

Finally, we show a reduction allowing to extend our decidability results from finite words to infinite ones, implying that the value 1 problem for probabilistic leaktight parity automata is decidable.

Introduction

Probabilistic automata. Rabin invented a very simple yet powerful model of probabilistic machine called probabilistic automata, which, quoting Rabin, "are a generalization of finite deterministic automata" [Rab63]. A probabilistic automaton has a finite set of states and reads input words from a finite alphabet. The computation starts from the initial state and consists in reading the input word sequentially; the state is updated according to transition probabilities determined by the current state and the input letter. The probability to accept a finite input word is the probability that the computation ends in one of the final states.

Probabilistic automata, and more generally partially observable Markov decision processes and stochastic games, are a widely studied model of probabilistic machines used in many fields like software verification [BBG12, CDHR07], image processing [CK97], computational biology [DEKM99] and speech processing [Moh97]. As a consequence, it is crucial to understand which decision problems are algorithmically tractable for probabilistic automata. From a language-theoretic perspective, several algorithmic properties of probabilistic automata are known: while language emptiness is undecidable [Paz71, Ber74, GO10], functional equivalence is decidable [Sch61, Tze92] as well as other properties [CMRR08].

Our initial motivation for this work comes from control and game theory: we aim at solving algorithmic questions about partially observable Markov decision processes and stochastic games. For this reason, we consider probabilistic automata as machines controlled by a blind controller, who is in charge of choosing the sequence of input letters in order to maximize the acceptance probability. While in a fully observable Markov decision process the controller can observe the current state of the process to choose adequately the next input letter, a blind controller does not observe anything and its choice depends only on the number of letters already chosen. In other words, the strategy of a blind controller is an input word of the automaton.

The value of a probabilistic automaton. With this game-theoretic interpretation in mind, we define the *value* of a probabilistic automaton as the supremum acceptance probability over all input words, and we would like to compute this value. Unfortunately, as a consequence of Paz undecidability result, the value of an automaton is not computable in general. However, the following decision problem was conjectured by Bertoni [Ber74] to be decidable:

Value 1 problem: *Given a probabilistic automaton, does it have value 1? In other words are there input words whose acceptance probability is arbitrarily close to 1?*

Recently, the second and fourth authors of the present paper proved that the value 1 problem is undecidable [GO10].

Our result. We introduce a new class of probabilistic automata, called *leaktight automata*, for which the value 1 problem is decidable. This subclass strictly subsumes all known subclasses of probabilistic automata sharing this decidability property and has good closure properties. Our algorithm to decide the value 1 problem computes in polynomial space a finite monoid whose elements are directed graphs and checks whether it contains a certain type of elements that are value 1 witnesses.

Related works. Introducing subclasses of probabilistic automata to cope with undecidability results has been a fruitful and lively topic recently. We discuss some of them here.

The first subclass which was introduced specifically to decide the value 1 problem are the $\sharp$-acyclic automata [GO10]. Later on, Chatterjee and Tracol [CT12] introduced structurally simple automata, which are probabilistic automata satisfying a structural property (related to the decomposition-separation theorem from probability theory), and proved that the value 1 problem is decidable for structurally simple automata. At the same time, a subset of the authors introduced leaktight automata, and proved a similar result. As we shall see, both $\sharp$-acyclic and structurally simple automata are leaktight, hence our results extend both [GO10] and [CT12].

Quite recently, Chadha, Sistla and Viswanathan introduced the subclass of hierarchical automata [CSV11], and showed that over infinite words, they recognize exactly the class of ω-regular languages. As we shall see, hierarchical automata are leaktight, hence as a consequence of our result, the value 1 problem is decidable for hierarchical automata.

Proof techniques. Our proof techniques totally depart from the ones used in [CSV11, CT12, GO10]. We make use of algebraic techniques and in particular Simon's factorization forest theorem, which was used successfully to prove the decidability of the boundedness problem for distance automata [Sim94], and extended models as desert automata and B-automata [Kir05, Col09]

Outline. Basic definitions are given in Section 1.

In Section 2, we introduce the Markov monoid and the Markov monoid algorithm for the value 1 problem; since the problem is in general undecidable, the algorithm is incomplete: a positive answer implies that the automaton has value 1, but a negative answer gives no guarantee.

In Section 3, we define the class of leaktight automata and show that the leaktight property is a sufficient condition for this algorithm to be complete; in particular, this implies that the value 1 problem is decidable for leaktight automata.

In Section 4, we show that the Markov monoid algorithm runs in polynomial space, and obtain as a corollary that the value 1 problem for leaktight automata is PSPACE-complete. Furthermore, we extend the Markov monoid algorithm to check at the same time whether an automaton is leaktight and whether in such case it has value 1.

In Section 5, we further investigate the class of leaktight automata: we provide examples of leaktight automata and show that all subclasses of probabilistic automata whose value 1 problem is known to be decidable are leaktight.

In Section 6, we give a general theorem allowing to extend the decidability results from finite words to infinite words.

1. DEFINITIONS

1.1. **Probabilistic automata.** We fix A a finite alphabet. A (finite) word u is a (possibly empty) sequence of letters $u = a_0 a_1 \cdots a_{n-1}$, the set of finite words is denoted by A^*. For $i \leq j$ we denote by $u[i, j]$ the subword $a_i \cdots a_{j-1}$, and $u_{<p} = u[0, p] = a_0 a_1 \cdots a_{p-1}$.

Let Q be a finite set of states. A probability distribution over Q is a function $\delta : Q \to [0, 1]$ such that $\sum_{q \in Q} \delta(q) = 1$; we often see δ as a row vector of size $|Q|$. We denote by $\frac{1}{3} \cdot q + \frac{2}{3} \cdot q'$ the distribution that picks q with probability $\frac{1}{3}$ and q' with probability $\frac{2}{3}$, and by q the trivial distribution picking q with probability 1. For a subset R of states, the uniform distribution over R picks each state in R with probability $\frac{1}{|R|}$. The support of a distribution

δ is the set of states picked with positive probability, *i.e.* $\mathrm{Supp}(\delta) = \{q \in Q \mid \delta(q) > 0\}$. Finally, the set of probability distributions over Q is $\mathcal{D}(Q)$.

Definition 1.1 (Probabilistic automaton). *A tuple $\mathcal{A} = (Q, q_0, \Delta, F)$ represents a probabilistic automaton, where Q is a finite set of states, $q_0 \in Q$ is the initial state, Δ defines the transitions and $F \subseteq Q$ is the set of accepting states.*

The transitions of a probabilistic automaton are given by a function $\Delta : Q \times A \to \mathcal{D}(Q)$, where $\Delta(q, a)$ is the probability distribution obtained by reading the letter a from the state q. The function Δ induces the function $\Delta' : \mathcal{D}(Q) \times A \to \mathcal{D}(Q)$, where $\Delta'(\delta, a) = \sum_{q \in Q} \delta(q) \cdot \Delta(q, a)$. Going further, Δ naturally extends to $\Delta^* : \mathcal{D}(Q) \times A^* \to \mathcal{D}(Q)$ by induction: for a letter $a \in A$, we set $\Delta^*(\delta, a) = \Delta'(\delta, a)$, and for an input word $u = av$, we set $\Delta^*(\delta, u) = \Delta^*(\Delta'(\delta, a), v)$. Intuitively, $\Delta^*(\delta, u)$ is the probability distribution obtained by reading the word u starting at the initial probability distribution δ. From now on, we will make no difference between Δ, Δ' and Δ^*, and denote the three of them by Δ.

We denote by $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t)$ the probability to go from state s to state t reading u on the automaton $\mathcal{A}$, *i.e.* $\Delta(s, u)(t)$. Then $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} T)$ is defined as $\sum_{t \in T} \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t)$. Finally, the *acceptance probability* of a word $u \in A^*$ by $\mathcal{A}$ is $\mathbb{P}_{\mathcal{A}}(q_0 \xrightarrow{u} F)$, which we denote by $\mathbb{P}_{\mathcal{A}}(u)$.

For computational purposes, we assume that each value is a rational number given by two integers in binary decomposition.

Definition 1.2 (Value). *The* value *of a probabilistic automaton $\mathcal{A}$, denoted by $val(\mathcal{A})$, is the supremum acceptance probability over all input words:*

$$val(\mathcal{A}) = \sup_{u \in A^*} \mathbb{P}_{\mathcal{A}}(u). \tag{1.1}$$

1.2. **The value 1 problem.** We are interested in the following decision problem:

Problem 1.3 (Value 1 Problem). *Given a probabilistic automaton $\mathcal{A}$, decide whether $val(\mathcal{A}) = 1$.*

The value 1 problem can be reformulated using the notion of *isolated cut-point* introduced by Rabin in his seminal paper [Rab63]: an automaton has value 1 if and only if the cut-point 1 is *not* isolated.
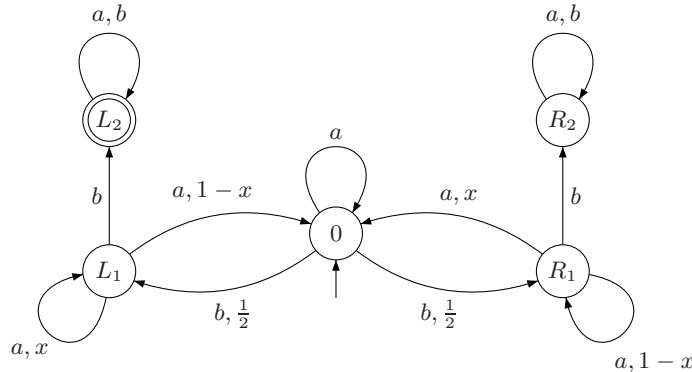

Figure 1: This automaton has value 1 if and only if $x > \frac{1}{2}$.

The automaton depicted on figure 1 has value 1 if and only if $x > \frac{1}{2}$ (a similar example appears in [BBG12]). The input alphabet is $A = \{a, b\}$, the initial state is the central state 0 and the unique final state is L_2.

We describe the behaviour of this automaton. After reading one b, the distribution is uniform over L_1, R_1. To reach L_2, one needs to read a b from the state L_1, but on the right-hand side this leads to the non-accepting absorbing state R_2. In order to maximize the probability to reach L_2, one tries to "tip the scales" to the left.

If $x \leq \frac{1}{2}$, there is no hope to achieve this: reading a letter a gives more chance to stay in R_1 than in L_1 thus all words are accepted with probability at most $\frac{1}{2}$, and $\text{val}(\mathcal{A}) = \frac{1}{2}$.

However, if $x > \frac{1}{2}$ then we show that $\mathcal{A}$ has value 1.

We have:

$$\mathbb{P}_\mathcal{A}(0 \xrightarrow{ba^n} L_1) = \frac{1}{2} \cdot x^n \qquad \text{and} \qquad \mathbb{P}_\mathcal{A}(0 \xrightarrow{ba^n} R_1) = \frac{1}{2} \cdot (1 - x)^n$$

We fix an integer N and analyse the action of reading $(ba^n)^N \cdot b$: there are N "rounds", each of them corresponding to reading ba^n from 0. In a round, there are three outcomes: winning (that is, remaining in L_1) with probability $p_n = \frac{1}{2} \cdot x^n$, losing (that is, remaining in R_2) with probability $q_n = \frac{1}{2} \cdot (1 - x)^n$, or going to the next round (that is, reaching 0) with probability $1 - (p_n + q_n)$. If a round is won or lost, then the next b leads to an accepting or rejecting sink; otherwise it goes on to the next round, for N rounds. Hence:

$$\begin{aligned}
\mathbb{P}_\mathcal{A}((ba^n)^N \cdot b) &= \sum_{k=1}^{N} (1 - (p_n + q_n))^{k-1} \cdot p_n \\
&= p_n \cdot \frac{1 - (1 - (p_n + q_n))^N}{1 - (1 - (p_n + q_n))} \\
&= \frac{1}{1 + \frac{q_n}{p_n}} \cdot \left(1 - (1 - (p_n + q_n))^N\right)
\end{aligned}$$

We now set $N = 2^n$. A simple calculation shows that the sequence $((1 - (p_n + q_n))^{2^n})_{n \in \mathbb{N}}$ converges to 0 as n goes to infinity. Furthermore, if $x > \frac{1}{2}$ then $\frac{1-x}{x} < 1$, so $\frac{q_n}{p_n} = \left(\frac{1-x}{x}\right)^n$ converges to 0 as n goes to infinity. It follows that the acceptance probability converges to 1 as n goes to infinity. Consequently:

$$\lim_n \mathbb{P}_\mathcal{A}((ba^n)^{2^n} \cdot b) = 1.$$

This example witnesses two surprising phenomena:
- the value is discontinuous with respect to the transition probabilities, as for $x = \frac{1}{2}$ the value is $\frac{1}{2}$, and for $x > \frac{1}{2}$ the value is 1;
- the sequence of words $((ba^n)^{2^n} \cdot b)_{n \in \mathbb{N}}$ witnessing the value 1 involves two convergence speeds: indeed, the words $a^n b$ are repeated an exponential number of times, namely 2^n. One can show that repeating only n times does not lead to words accepted with arbitrarily high probability.

1.3. **Recurrent states and idempotent words.** We fix $\mathcal{A}$ a probabilistic automaton, and define two main notions: recurrent states and idempotent words.

Definition 1.4 (Induced Markov chain)**.** *Let u be a finite word, it induces a Markov chain* $\mathcal{M}_{\mathcal{A},u}$ *whose state space is Q and transition matrix $M_{\mathcal{A},u}$ is defined by:*

$$M_{\mathcal{A},u}(s, t) = \mathbb{P}_\mathcal{A}(s \xrightarrow{u} t).$$

We rely on the classical notion of recurrent states in Markov chains.

Definition 1.5 (Recurrent state). *A state s is u-recurrent if it is recurrent in $\mathcal{M}_{\mathcal{A},u}$.*

A finite word u is *idempotent* if reading once or twice the word u does not change qualitatively the transition probabilities.

Definition 1.6 (Idempotent word). *A Markov chain is idempotent if its transition matrix M satisfies that for all states s, t:*

$$M(s,t) > 0 \iff M^2(s,t) > 0.$$

A finite word u is idempotent if $\mathcal{M}_{\mathcal{A},u}$ is idempotent.

In the case of idempotent words, recurrence of a state is easily characterized, relying on simple graph-theoretical arguments:

Lemma 1.7. *Let u be an idempotent word. A state s is u-recurrent if and only if for all states t we have:*

$$\mathcal{M}_{\mathcal{A},u}(s,t) > 0 \implies \mathcal{M}_{\mathcal{A},u}(t,s) > 0.$$

2. An (incomplete) algorithm for the value 1 problem

In this section, we present an algebraic algorithm for the value 1 problem, called the Markov monoid algorithm. Since the problem is undecidable, this algorithm does not solve the problem on all instances; we will show that it is *correct*, *i.e.* if it answers that an automaton has value 1, then the automaton does have value 1, but not *complete*, *i.e.* the converse does not hold. In the next section, we shall show that this algorithm is *complete* for the class of leaktight automata.

2.1. **The Markov monoid algorithm.** Our algorithm for the value 1 problem computes iteratively a set $\mathcal{G}$ of directed graphs called limit-words. Each limit-word is meant to represent the asymptotic effect of a sequence of input words, and some particular limit-words can witness that the automaton has value 1.

Definition 2.1 (Limit-word). *A limit-word is a function $\mathbf{u} : Q^2 \to \{0, 1\}$, such that for all states s, there exists a state t such that $\mathbf{u}(s, t) = 1$.*

In proofs and examples, we will adopt either of the two equivalent views for limit-words: graphs over the set Q or square matrices over $Q \times Q$.

We now explain the algorithm in detail. For the remainder of this section, we fix $\mathcal{A}$ a probabilistic automaton. Initially, $\mathcal{G}$ only contains those limit-words $\mathbf{a}$ that are induced by input letters $a \in A$:

$$\forall s, t \in Q, \ (\mathbf{a}(s,t) = 1 \iff \mathbb{P}_{\mathcal{A}}(s \xrightarrow{a} t) > 0),$$

plus the limit-word $\mathbf{1}$ which is induced by the empty word:

$$\forall s, t \in Q, \ (\mathbf{1}(s,t) = 1 \iff s = t).$$

The algorithm repeatedly adds new limit-words to $\mathcal{G}$. There are two ways for that: concatenating two limit-words or iterating an idempotent limit-word.

ALGORITHM 1: The Markov monoid algorithm.

Data: A probabilistic automaton.

$\mathcal{G} \leftarrow \{\mathbf{a} \mid a \in A\} \cup \{\mathbf{1}\}$.

repeat

   | **if** *there is* $\mathbf{u}, \mathbf{v} \in \mathcal{G}$ *such that* $\mathbf{u} \cdot \mathbf{v} \notin \mathcal{G}$ **then**

   | | add $\mathbf{u} \cdot \mathbf{v}$ to $\mathcal{G}$

   | **end**

   | **if** *there is* $\mathbf{u} \in \mathcal{G}$ *such that* $\mathbf{u}$ *is idempotent and* $\mathbf{u}^\sharp \notin \mathcal{G}$ **then**

   | | add $\mathbf{u}^\sharp$ to $\mathcal{G}$

   | **end**

until *there is nothing to add*;

if *there is a value* 1 *witness in* $\mathcal{G}$ **then**

   | return true;

else

   | return false;

end

Concatenation of two limit-words The *concatenation* of two limit-words $\mathbf{u}$ and $\mathbf{v}$ is the limit-word $\mathbf{u} \cdot \mathbf{v}$ such that:

$$(\mathbf{u} \cdot \mathbf{v})(s, t) = 1 \iff \exists q \in Q, \ \mathbf{u}(s, q) = 1 \text{ and } \mathbf{v}(q, t) = 1.$$

In other words, concatenation corresponds to the multiplication of matrices with coefficients in the boolean semiring $(\{0, 1\}, \vee, \wedge)$. Intuitively, the concatenation of two limit-words corresponds to the concatenation of two sequences $(u_n)_{n \in \mathbb{N}}$ and $(v_n)_{n \in \mathbb{N}}$ of input words into the sequence $(u_n \cdot v_n)_{n \in \mathbb{N}}$.

We say that a limit-word $\mathbf{u}$ is idempotent if $\mathbf{u} \cdot \mathbf{u} = \mathbf{u}$. The following lemma gives simple properties of idempotent limit-words.

Lemma 2.2. *For all limit-words* $\mathbf{u}$:

- *the limit-word* $\mathbf{u}^{|Q|!}$ *is idempotent,*
- *if* $\mathbf{u}$ *is idempotent, then for all states* $r \in Q$, *there exists a state* $r' \in Q$ *such that* $\mathbf{u}(r, r') = 1$ *and* r' *is* $\mathbf{u}$-*recurrent.*

The proof is omitted and relies on simple graph-theoretical arguments.

Iteration of an idempotent limit-word The *iteration* $\mathbf{u}^\sharp$ of a limit-word $\mathbf{u}$ is only defined when $\mathbf{u}$ is idempotent. It relies on the notion of $\mathbf{u}$-recurrent state.

Definition 2.3 ($\mathbf{u}$-recurrence)**.** *Let* $\mathbf{u}$ *be an idempotent limit-word. A state* s *is* $\mathbf{u}$-*recurrent if for all states* t, *we have:*

$$\mathbf{u}(s, t) = 1 \implies \mathbf{u}(t, s) = 1.$$

Note that this echoes Lemma 1.7. The *iterated limit-word* $\mathbf{u}^\sharp$ removes from $\mathbf{u}$ any edge that does not lead to a recurrent state:

$$\mathbf{u}^\sharp(s, t) = 1 \iff \mathbf{u}(s, t) = 1 \text{ and } t \text{ is } \mathbf{u}\text{-recurrent.}$$

Intuitively, if a limit-word $\mathbf{u}$ represents a sequence $(u_n)_{n \in \mathbb{N}}$ then its iteration $\mathbf{u}^\sharp$ represents the sequence $\left(u_n^{f(n)} \right)_{n \in \mathbb{N}}$ for some increasing function $f : \mathbb{N} \to \mathbb{N}$.

2.2. **The Markov monoid and value** 1 **witnesses.** The set of limit-words $\mathcal{G}$ computed by Algorithm 1 is called the Markov monoid.

Definition 2.4 (Markov monoid). *The Markov monoid associated with $\mathcal{A}$ is the smallest set of limit-words containing $\{\mathbf{a} \mid a \in A\} \cup \{\mathbf{1}\}$ and closed under concatenation and iteration.*

Two key properties, *consistency* and *completeness*, ensure that the limit-words of the Markov monoid reflect exactly every possible asymptotic effect of a sequence of input words.

Definition 2.5 (Reification). *A sequence $(u_n)_{n \in \mathbb{N}}$ of words reifies a limit-word $\mathbf{u}$ if for all states s, t, $(\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t))_{n \in \mathbb{N}}$ converges and:*

$$\mathbf{u}(s,t) = 1 \iff \lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) > 0. \tag{2.1}$$

Note that if $(u_n)_{n \in \mathbb{N}}$ reifies $\mathbf{u}$, then any subsequence of $(u_n)_{n \in \mathbb{N}}$ also does. We will use this simple observation several times.

Definition 2.6 (Consistency). *A set of limit-words $\mathcal{G}$ is* consistent *with $\mathcal{A}$ if for every limit-word $\mathbf{u} \in \mathcal{G}$, there exists a sequence of input words $(u_n)_{n \in \mathbb{N}}$ which reifies $\mathbf{u}$.*

Definition 2.7 (Completeness). *A set of limit-words $\mathcal{G}$ is* complete *for $\mathcal{A}$ if for each sequence of input words $(u_n)_{n \in \mathbb{N}}$, there exists $\mathbf{u} \in \mathcal{G}$ such that for all states $s, t \in Q$:*

$$\limsup_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) = 0 \implies \mathbf{u}(s,t) = 0. \tag{2.2}$$

Limit-words are useful to decide the value 1 problem because some of these are witnesses that the automaton has value 1.

Definition 2.8 (Value 1 witness). *A value 1 witness is a limit-word $\mathbf{u}$ such that for all states t:*

$$\mathbf{u}(q_0, t) = 1 \implies t \in F, \tag{2.3}$$

where q_0 is the initial state of the automaton.

Thanks to value 1 witnesses, the answer to the value 1 problem can be read in a consistent and complete set of limit-words:

Lemma 2.9 (A criterion for value 1). *If $\mathcal{G}$ is consistent with $\mathcal{A}$ and complete for $\mathcal{A}$, then $\mathcal{A}$ has value 1 if and only if $\mathcal{G}$ contains a value 1 witness.*
    Specifically:
- *If $\mathcal{G}$ is consistent with $\mathcal{A}$ and contains a value 1 witness, then $\mathcal{A}$ has value 1,*
- *If $\mathcal{G}$ is complete for $\mathcal{A}$ and $\mathcal{A}$ has value 1, then $\mathcal{A}$ contains a value 1 witness.*

Proof. We prove the first item. Assume that $\mathcal{G}$ is consistent with $\mathcal{A}$ and contains a value 1 witness $\mathbf{u}$. Since $\mathcal{G}$ is consistent, there exists a sequence $(u_n)_{n \in \mathbb{N}}$ reifying $\mathbf{u}$. It follows from (2.1) and (2.3) that for $t \notin F$, we have $\lim_n \mathbb{P}_{\mathcal{A}}(q_0 \xrightarrow{u_n} t) = 0$. Thus $\lim_n \mathbb{P}_{\mathcal{A}}(u_n) = \sum_{t \in F} \lim_n \mathbb{P}_{\mathcal{A}}(q_0 \xrightarrow{u_n} t) = 1$, so $\mathcal{A}$ has value 1.
    We now prove the second item. Assume that $\mathcal{G}$ is complete for $\mathcal{A}$ and that $\mathcal{A}$ has value 1. Then there exists a sequence of words $(u_n)_{n \in \mathbb{N}}$ such that $\lim_n \mathbb{P}_{\mathcal{A}}(u_n) = 1$, *i.e.* $\lim_n \sum_{t \in F} \mathbb{P}_{\mathcal{A}}(q_0 \xrightarrow{u_n} t) = 1$. Since for all $n \in \mathbb{N}$, we have $\sum_{q \in Q} \mathbb{P}_{\mathcal{A}}(q_0 \xrightarrow{u_n} q) = 1$, then for all $t \notin F$, $\limsup_n \mathbb{P}_{\mathcal{A}}(q_0 \xrightarrow{u_n} t) = 0$. Since $\mathcal{G}$ is complete, there exists a limit-word $\mathbf{u}$ such that (2.2) holds. Then $\mathbf{u}$ is a value 1 witness: let $t \in Q$ such that $\mathbf{u}(q_0, t) = 1$, then according to (2.2), $\limsup_n \mathbb{P}_{\mathcal{A}}(q_0 \xrightarrow{u_n} t) > 0$, hence $t \in F$. $\square$

2.3. **Correctness of the Markov monoid algorithm.**

Theorem 2.10. *The Markov monoid associated with $\mathcal{A}$ is consistent.*

This implies that if the Markov monoid algorithm outputs "true", then for sure the input automaton has value 1. This positive result holds for every automaton (leaktight or not).

To prove Theorem 2.10, recall that the Markov monoid is the smallest set of limit-words containing $\{\mathbf{a} \mid a \in A\} \cup \{\mathbf{1}\}$ and closed under concatenation and iteration, hence it suffices to prove that the initial elements form a consistent set, and the closure under the two operations.

First, $\mathbf{a}$ is reified by the constant sequence $(a)_{n \in \mathbb{N}}$, and $\mathbf{1}$ by the constant sequence $(\varepsilon)_{n \in \mathbb{N}}$. We state the closure under the two operations in the following proposition:

Proposition 2.11. *Let $(u_n)_{n \in \mathbb{N}}$ and $(v_n)_{n \in \mathbb{N}}$ be two sequences that reify the limit-words $\mathbf{u}$ and $\mathbf{v}$ respectively. Then:*

(1) the sequence of words $(u_n \cdot v_n)_{n \in \mathbb{N}}$ reifies $\mathbf{u} \cdot \mathbf{v}$,

(2) if $\mathbf{u}$ is idempotent, then there exists an increasing function $f : \mathbb{N} \to \mathbb{N}$ such that for all increasing functions $g : \mathbb{N} \to \mathbb{N}$ satisfying $g \geq f$, the sequence $\left(u_{g(n)}^n \right)_{n \in \mathbb{N}}$ reifies the limit-word $\mathbf{u}^\sharp$.

The statement about iteration is stronger than required: the existence of f such that $(u_{f(n)}^n)_{n \in \mathbb{N}}$ reifying the limit-word $\mathbf{u}^\sharp$ is enough to prove Theorem 2.10. However, we will use this stronger result later on (in Section 5.4).

Proof.

(1) Let $w_n = u_n \cdot v_n$. Then $(w_n)_{n \in \mathbb{N}}$ reifies $\mathbf{u} \cdot \mathbf{v}$, since:

$$\mathbb{P}_{\mathcal{A}}(s \xrightarrow{w_n} t) = \sum_{r \in Q} \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} r) \cdot \mathbb{P}_{\mathcal{A}}(r \xrightarrow{v_n} t).$$

(2) Consider the Markov chain $\mathcal{M}$ with state space Q and transition matrix M defined by $M(s,t) = \lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t)$. Since $(u_n)_{n \in \mathbb{N}}$ reifies $\mathbf{u}$, we have $\mathbf{u}(s,t) = 1$ if and only if $M(s,t) > 0$. First observe that since $\mathbf{u}$ is idempotent, the Markov chain $\mathcal{M}$ is aperiodic. According to standard results about finite Markov chains, this implies that the sequence of matrices $(M^k)_{k \in \mathbb{N}}$ has a limit which we denote by M^∞, satisfying the following:

$$\forall s,t \in Q, \; M^\infty(s,t) > 0 \implies t \text{ is recurrent in } \mathcal{M}. \tag{2.4}$$

By definition the sequence of matrices $(M_{\mathcal{A},u_n})_{n \in \mathbb{N}}$ converges to M. Since the matrix product operation is continuous, for every $k \in \mathbb{N}$, $\left(M_{\mathcal{A},u_n}^k \right)_{n \in \mathbb{N}}$ converges to M^k. So for every $k \geq 1$, there exists $N_k \in \mathbb{N}$ such that for all $p \geq N_k$, $||M^k - M_{\mathcal{A},u_p}^k||_\infty \leq \frac{1}{k}$. We define $f : \mathbb{N} \to \mathbb{N}$ by induction, so that $f(k)$ is the maximum of $f(k-1)+1$ and of N_k, ensuring that f is increasing. Then for any increasing function $g : \mathbb{N} \to \mathbb{N}$ satisfying $g \geq f$, the sequence of matrices $\left(M_{\mathcal{A},u_{g(n)}}^n \right)_{n \in \mathbb{N}}$ converges to M^∞. We

prove that $\left(u^n_{g(n)}\right)_{n\in\mathbb{N}}$ reifies $\mathbf{u}^\sharp$:

$$
\begin{aligned}
\mathbf{u}^\sharp(s,t) = 1 &\iff \mathbf{u}(s,t) = 1 \text{ and } t \text{ is } \mathbf{u}\text{-recurrent} \\
&\iff M(s,t) > 0 \text{ and } t \text{ is recurrent in } \mathcal{M} \\
&\iff M^\infty(s,t) > 0 \\
&\iff \lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u^n_{g(n)}} t) > 0,
\end{aligned}
$$

where the first equivalence is by definition of the iteration, the second holds because $(u_n)_{n\in\mathbb{N}}$ reifies $\mathbf{u}$, the third by definition of M^∞, and the fourth because $\left(M_{\mathcal{A},u^n_{g(n)}}\right)_{n\in\mathbb{N}}$ converges to M^∞.

This concludes the proof. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Note that completeness is not true in general; for instance, one can show that the Markov monoid of the automaton represented in figure 1 is not complete. The next section gives a sufficient condition for completeness: the leaktight property.

3. Decidability of the value 1 problem for leaktight automata

In this section we establish our main result:

Theorem 3.1. *The value 1 problem is decidable for leaktight automata.*

The definition of leaktight automata is given in the next subsection. For now (in this section), we are only interested in decidability issues; we will actually prove in Section 4 that the value 1 problem is PSPACE-complete for leaktight automata.

Note that as observed in the literature [BBG12, CSV13, Fij14], the value 1 problem for probabilistic automata over finite words is equivalent to the emptiness problem for probabilistic Büchi automata with positive semantics, hence we obtain the following corollary:

Corollary 3.2. *The emptiness problem is decidable for probabilistic Büchi leaktight automata with positive semantics.*

The following theorem proves that the Markov monoid of a leaktight automaton is complete; since it is always consistent, by Lemma 2.9, the Markov monoid algorithm solves the value 1 problem for leaktight automata.

Theorem 3.3. *If a probabilistic automaton is leaktight then its Markov monoid is complete.*

The remainder of this section is devoted to the proof of Theorem 3.3. We first define the leaktight property, and extend the Markov monoid. This extended version allows to state an algebraic characterization of the leaktight property. Then, the technical core of the proof relies on a subtle algebraic argument based on the existence of $\sharp$-factorization trees of bounded height [Sim90, Sim94, Col09, Tor11].

3.1. **Leaks.** The undecidability of the value 1 problem comes from the necessity to compare parallel convergence rates in order to track down vanishing probabilities. Comparing two convergence rates may require to compare the decimals of the rates up to an arbitrary precision, which in turn can encode a Post correspondence problem, hence the undecidability.

One of the phenomena that makes tracking vanishing probabilities difficult are *leaks*. A leak occurs in an automaton when a sequence of words turns a set of states $C \subseteq Q$ into a recurrence class C *on the long run*, but *on the short run*, some of the probability of the recurrence class is "leaking" to a *different* recurrence class.
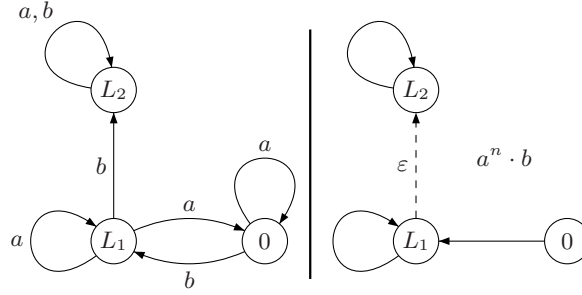

Figure 2: $(a^n \cdot b)_{n \in \mathbb{N}}$ is a leak from L_1 to L_2.

Such leaks occur in the automaton depicted in the left hand side of figure 2 with the input sequence $(a^n b)_{n \in \mathbb{N}}$. As n grows large, the probability to reach L_2 from L_1 while reading the input word $a^n b$ vanishes, thus the sets $\{L_1\}$ and $\{L_2\}$ are two different recurrence classes on the long run (*i.e.* asymptotically), however on the short run remains a small yet positive probability to reach L_2 from L_1.

The right hand side of figure 2 shows the asymptotic behaviour of reading $(a^n b)_{n \in \mathbb{N}}$.

Since the automaton in figure 1 contains two symmetric parts identical to figure 2, it features one leak on the left hand side and another in the right hand side. As a consequence, the real asymptotic behaviour is complex and depends on the compared speeds of these leaks.

An automaton without leak is called a leaktight automaton. In this section we prove that the value 1 problem is decidable when restricted to the subclass of leaktight automata.

The formal definition of a leak is as follows:

Definition 3.4 (Leak). *Let $(u_n)_{n \in \mathbb{N}}$ 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 assocaited Markov chain.*

The sequence $(u_n)_{n \in \mathbb{N}}$ is a leak if there exist $r, q \in Q$ such that the following three conditions hold:

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

Definition 3.5 (Leaktight automata). *A probabilistic automaton is leaktight if it has no leak.*

Several examples of leaktight automata are given in Section 5.

3.2. **The extended Markov monoid.** The existence of leaks can be decided by a slight extension of the Markov monoid algorithm which keeps track of strictly positive transition probabilities.

Definition 3.6 (Extended limit-word). *An* extended limit-word *is a couple* $(\mathbf{u}, \mathbf{u}_+)$ *of two limit-words, such that for all* $s, t \in Q$, *we have* $\mathbf{u}(s, t) = 1 \Longrightarrow \mathbf{u}_+(s, t) = 1$.

As for limit-words, extended limit-words can be seen either as graphs over the set Q, or couples of square matrices over $Q \times Q$. Such a graph has two different kind of edges: an edge (s, t) is "normal" if $\mathbf{u}(s, t) = 1$, and is a +-edge if $\mathbf{u}(s, t) = 0$ but $\mathbf{u}_+(s, t) = 1$.

We define the concatenation and iteration operations for extended limit-words. The *concatenation* of two extended limit-words $(\mathbf{u}, \mathbf{u}_+)$ and $(\mathbf{v}, \mathbf{v}_+)$ is the component-wise concatenation, *i.e.* $(\mathbf{u} \cdot \mathbf{v}, \mathbf{u}_+ \cdot \mathbf{v}_+)$. The *iteration* of an extended limit-word $(\mathbf{u}, \mathbf{u}_+)$ is only defined when it is idempotent (*i.e.* component-wise idempotent), by $(\mathbf{u}, \mathbf{u}_+)^\sharp = (\mathbf{u}^\sharp, \mathbf{u}_+)$.

Definition 3.7 (Extended Markov monoid). *The extended Markov monoid is the smallest set of extended limit-words containing* $\{(\mathbf{a}, \mathbf{a}) \mid a \in A\} \cup \{(\mathbf{1}, \mathbf{1})\}$ *and closed under concatenation and iteration.*

Note that if $(\mathbf{u}, \mathbf{u}_+)$ is in the extended Markov monoid, then $\mathbf{u}$ is in the Markov monoid.

The essential difference between the Markov monoid and its extended version is that the extension keeps track of those edges that are deleted by successive iteration operations. This serves two purposes: first, to characterize the leaktight property in algebraic terms, and second, to prove Theorem 3.3.

We state a consistency result for the extended Markov monoid, extending Theorem 2.10. The proofs of both these results are similar and given only once.

Lemma 3.8. *For each* $(\mathbf{u}, \mathbf{u}_+)$ *in the extended Markov monoid, there exists a sequence* $(u_n)_{n \in \mathbb{N}}$ *such that for all states* $s, t \in Q$, $(\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t))_{n \in \mathbb{N}}$ *converges and:*

$$\mathbf{u}(s, t) = 1 \iff \lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) > 0, \tag{3.1}$$

$$\text{for all } n \in \mathbb{N}, \ \left(\mathbf{u}_+(s, t) = 1 \iff \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) > 0 \right). \tag{3.2}$$

3.3. **Leak witnesses.**

Definition 3.9 (Leak witness). *An idempotent extended limit-word* $(\mathbf{u}, \mathbf{u}_+)$ *is a* leak witness *if there exist* $r, q \in Q$ *such that the following three conditions hold:*

    (1) r *and* q *are* $\mathbf{u}$-*recurrent,*
    (2) $\mathbf{u}(r, q) = 0$,
    (3) $\mathbf{u}_+(r, q) = 1$.

Lemma 3.10. *If a probabilistic automaton is leaktight, then its extended Markov monoid does not contain any leak witness.*

Proof. Suppose that there is a leak witness $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid: $\mathbf{u}$ and $\mathbf{u}_+$ are idempotent and there exists $r, q \in Q$ such that r and q are $\mathbf{u}$-recurrent, $\mathbf{u}(r, q) = 0$ and $\mathbf{u}_+(r, q) = 1$. We prove that there exists a leak.

Thanks to Lemma 3.8, there exists a sequence $(u_n)_{n \in \mathbb{N}}$ satisfying (3.1) and (3.2). Note that since $\mathbf{u}_+$ is idempotent, (3.2) implies that for all $n \in \mathbb{N}$, u_n is idempotent.

Consider the Markov chain $\mathcal{M}$ with state space Q and transition matrix M defined by $M(s,t) = \lim_n \mathbb{P}_\mathcal{A}(s \xrightarrow{u_n} t)$. $\mathcal{M}$ is idempotent since $\mathbf{u}$ is idempotent and thanks to (3.1).

We show that $(u_n)_{n \in \mathbb{N}}$ is a leak. There are three conditions to be met.

First, r and q are recurrent in $\mathcal{M}$: this follows from (3.1) and the fact that r and q are $\mathbf{u}$-recurrent. Second, $\lim_n \mathbb{P}_\mathcal{A}(r \xrightarrow{u_n} q) = 0$: this follows from (3.1) and the fact that $\mathbf{u}(r,q) = 0$. Third, for all $n \in \mathbb{N}$, $\mathbb{P}_\mathcal{A}(r \xrightarrow{u_n} q) > 0$: this follows from (3.2) and the fact that $\mathbf{u}_+(r,q) = 1$. $\qquad\qquad\square$

As we will show in the next section, the converse of Lemma 3.10 is also true, which gives an algebraic characterization of the leaktight property using the extended Markov monoid. However, the proof of the converse implication is more involved and requires the lower bound lemma (Lemma 3.15), which is the object of the next subsection.

3.4. **Stabilization monoids and $\sharp$-factorization trees.** We now introduce the technical material required to state and prove the lower bound lemma. The key notions here are stabilization monoids and $\sharp$-factorization trees.

Factorization trees for monoids have been introduced by Simon [Sim90]. Roughly speaking, Simon's factorization theorem states that given a morphism $\phi : A^* \to M$ from the set of finite words over A to a finite monoid M, the following holds: for all words u, the computation of $\phi(u)$ can be factorized in a tree whose depth is bounded independently of the length of the word.

Simon later developed the notion of decomposition trees to solve the limitedness problem for distance automata [Sim94]. To this end, he defined an iteration operation $\sharp$ for monoids over the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +)$. Then Kirsten extended this technique to desert automata and the nested distance desert automata [Kir05]. After him, Colcombet generalized this approach by defining stabilization monoids [Col09], which are monoids equipped with an iteration operation, and proved the existence of $\sharp$-factorization trees of bounded depth. The formal definition is as follows:

Definition 3.11 (Stabilization monoid). *A stabilization monoid $(M, \cdot, \sharp)$ is a finite monoid $(M, \cdot)$ equipped with an iteration operation $\sharp : E(M) \to E(M)$, where $E(M)$ is the set of idempotents of M, such that:*

$$(a \cdot b)^\sharp \cdot a = a \cdot (b \cdot a)^\sharp \qquad \text{for } a \cdot b \in E(M) \text{ and } b \cdot a \in E(M), \qquad (3.3)$$

$$(e^\sharp)^\sharp = e^\sharp \qquad\qquad \text{for } e \in E(M), \qquad (3.4)$$

$$e^\sharp \cdot e = e^\sharp \qquad\qquad \text{for } e \in E(M). \qquad (3.5)$$

Lemma 3.12. *The extended Markov monoid is a stabilization monoid.*

Proof. To start with, the extended Markov monoid is a monoid for the concatenation: $\mathbf{1}$ is the neutral element, and the concatenation is associative.

Now, let us prove the three properties required for the iteration operation $\sharp$.

Proof of (3.3). Let $(\mathbf{u}, \mathbf{u}_+), (\mathbf{v}, \mathbf{v}_+)$ such that $(\mathbf{u} \cdot \mathbf{v}, \mathbf{u}_+ \cdot \mathbf{v}_+)$ and $(\mathbf{v} \cdot \mathbf{u}, \mathbf{v}_+ \cdot \mathbf{u}_+)$ are idempotent. By definition $((\mathbf{u}, \mathbf{u}_+) \cdot (\mathbf{v}, \mathbf{v}_+))^\sharp \cdot (\mathbf{u}, \mathbf{u}_+)$ is equal to $((\mathbf{u} \cdot \mathbf{v})^\sharp \cdot \mathbf{u}, \mathbf{u}_+ \cdot \mathbf{v}_+ \cdot \mathbf{u}_+)$, and $(\mathbf{u}, \mathbf{u}_+) \cdot ((\mathbf{v}, \mathbf{v}_+) \cdot (\mathbf{u}, \mathbf{u}_+))^\sharp$ to $(\mathbf{u} \cdot (\mathbf{v} \cdot \mathbf{u})^\sharp, \mathbf{u}_+ \cdot \mathbf{v}_+ \cdot \mathbf{u}_+)$. Let $s, t \in Q$, we have the following equivalence: $((\mathbf{u} \cdot \mathbf{v})^\sharp \cdot \mathbf{u})(s, t) = 1$ if and only if:

there exists $r, q \in Q$, $\mathbf{u}(s, r) = 1 \wedge \mathbf{v}(r, q) = 1 \wedge \mathbf{u}(q, t) = 1 \wedge q$ is $(\mathbf{u} \cdot \mathbf{v})$-recurrent, (3.6)

and similarly, $\left(\mathbf{u} \cdot (\mathbf{v} \cdot \mathbf{u})^\sharp\right)(s,t) = 1$ if and only if:

there exists $r, q \in Q$, $\mathbf{u}(s,r) = 1 \wedge \mathbf{v}(r,q) = 1 \wedge \mathbf{u}(q,t) = 1 \wedge t$ is $(\mathbf{v} \cdot \mathbf{u})$-recurrent.   (3.7)

We show that (3.6) and (3.7) are equivalent. Assume (3.6), and prove that t is $(\mathbf{v} \cdot \mathbf{u})$-recurrent. Let $p \in Q$ such that $(\mathbf{v} \cdot \mathbf{u})(t,p) = 1$. Since $\mathbf{v}$ is a limit-word, there exists $\ell \in Q$ such that $\mathbf{v}(p,\ell) = 1$. Observe that $\mathbf{u}(q,t) = 1$, $(\mathbf{v} \cdot \mathbf{u})(t,p) = 1$ and $\mathbf{v}(p,\ell) = 1$, so $(\mathbf{u} \cdot \mathbf{v})^2(q,\ell) = 1$. As $\mathbf{u} \cdot \mathbf{v}$ is idempotent, this implies $(\mathbf{u} \cdot \mathbf{v})(q,\ell) = 1$. Since q is $(\mathbf{u} \cdot \mathbf{v})$-recurrent, we have $(\mathbf{u} \cdot \mathbf{v})(\ell,q) = 1$. Altogether, $\mathbf{v}(p,\ell) = 1$, $(\mathbf{u} \cdot \mathbf{v})(\ell,q) = 1$ and $\mathbf{u}(q,t) = 1$ imply that $(\mathbf{v} \cdot \mathbf{u})^2(p,t) = 1$. As $\mathbf{v} \cdot \mathbf{u}$ is idempotent, this implies $(\mathbf{v} \cdot \mathbf{u})(p,t) = 1$, so t is $(\mathbf{v} \cdot \mathbf{u})$-recurrent, and (3.7) is proved. Conversely, assume (3.7), and prove that q is $(\mathbf{u} \cdot \mathbf{v})$-recurrent. Note that from $\mathbf{v}(r,q) = 1$, $\mathbf{u}(q,t) = 1$ and the fact that t is $(\mathbf{v} \cdot \mathbf{u})$-recurrent, we obtain that $(\mathbf{v} \cdot \mathbf{u})(t,r) = 1$. Let $p \in Q$ such that $(\mathbf{u} \cdot \mathbf{v})(q,p) = 1$. Since $\mathbf{u}$ is a limit-word, there exists $\ell \in Q$ such that $\mathbf{u}(p,\ell) = 1$. Observe that $\mathbf{v}(r,q) = 1$, $(\mathbf{u} \cdot \mathbf{v})(q,p) = 1$ and $\mathbf{u}(p,\ell) = 1$, so $(\mathbf{v} \cdot \mathbf{u})^2(r,\ell) = 1$, and with $(\mathbf{v} \cdot \mathbf{u})(t,r) = 1$ this implies $(\mathbf{v} \cdot \mathbf{u})^3(t,\ell) = 1$. As $\mathbf{v} \cdot \mathbf{u}$ is idempotent, this implies $(\mathbf{v} \cdot \mathbf{u})(t,\ell) = 1$. Since t is $(\mathbf{v} \cdot \mathbf{u})$-recurrent, we have $(\mathbf{v} \cdot \mathbf{u})(\ell,t) = 1$. Altogether, $\mathbf{u}(p,\ell) = 1$, $(\mathbf{v} \cdot \mathbf{u})(\ell,t) = 1$, $(\mathbf{v} \cdot \mathbf{u})(t,r) = 1$ and $\mathbf{v}(r,q) = 1$ imply that $(\mathbf{u} \cdot \mathbf{v})^3(p,q) = 1$. As $\mathbf{u} \cdot \mathbf{v}$ is idempotent, this implies $(\mathbf{u} \cdot \mathbf{v})(p,q) = 1$, so q is $(\mathbf{u} \cdot \mathbf{v})$-recurrent, and (3.6) is proved. The property (3.3) follows.

Proof of (3.4). This boils down to proving $(\mathbf{u}^\sharp)^\sharp = \mathbf{u}^\sharp$. This is clear from the definition of $\mathbf{u}^\sharp$, since the notions of $\mathbf{u}$-recurrence and $\mathbf{u}^\sharp$-recurrence coincide.

Proof of (3.5). This boils down to proving $\mathbf{u}^\sharp \cdot \mathbf{u} = \mathbf{u}^\sharp$. It follows from the observation that if $r \in Q$ is $\mathbf{u}$-recurrent and $\mathbf{u}(r,t) = 1$, then t is $\mathbf{u}$-recurrent (under the assumption that $\mathbf{u}$ is idempotent).   □

Definition 3.13. *Let A be a finite alphabet, $(M,\cdot,\sharp)$ a stabilization monoid and $\phi : A^* \to M$ a morphism into the submonoid $(M,\cdot)$. A $\sharp$-factorization tree of a word $u \in A^*$ is a finite unranked ordered tree, whose nodes have labels in $A^* \times M$ and such that:*

  i) *the root is labelled by $(u,\mathbf{u})$, for some $\mathbf{u} \in M$,*
  ii) *every internal node with two children (called* concatenation *nodes) labelled by $(u_1,\mathbf{u}_1)$ and $(u_2,\mathbf{u}_2)$ is labelled by $(u_1 \cdot u_2, \mathbf{u}_1 \cdot \mathbf{u}_2)$,*
  iii) *every internal node with three or more children (called* iteration *nodes) is labelled by $(u_1 \ldots u_n, \mathbf{e}^\sharp)$ for some $\mathbf{e} \in E(M)$, and its children are labelled by $(u_1,\mathbf{e}),\ldots,(u_n,\mathbf{e})$.*
  iv) *every leaf is labelled by $(a,\mathbf{a})$ where a is a letter, or $(\varepsilon,\mathbf{1})$.*

Note that in a factorization tree, the second label is not always the image of the first component under ϕ; indeed, it is an element of the stabilization monoid $(M,\cdot,\sharp)$ whereas the image of a finite word under ϕ is an element of the submonoid $(M,\cdot)$. However, the projection of second label into this submonoid (which consists in ignoring the operation $\sharp$) is indeed the image of the first component under ϕ.

The following theorem was stated for the tropical semiring in [Sim94], and generalized in [Col09]. A simple proof can be found in [Tor11].

Theorem 3.14. *Let A be a finite alphabet, $(M,\cdot,\sharp)$ a stabilization monoid and $\phi : A^* \to M$ a morphism into the submonoid $(M,\cdot)$. Every word $u \in A^*$ has a $\sharp$-factorization tree whose depth is less than $3 \cdot |M|$.*

3.5. **The lower bound lemma.** We are ready to state and prove the lower bound lemma, which is the central argument in the proof of completeness of leaktight Markov monoids.

Lemma 3.15 (Lower bound lemma). *Let $\mathcal{A}$ be a probabilistic automaton whose extended Markov monoid contains no leak witness. Let $p_{\min}$ the smallest non-zero transition probability of $\mathcal{A}$. Then for all words $u \in A^*$, there exists $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid such that, for all states s, t:*

$$\mathbf{u}_+(s,t) = 1 \iff \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) > 0, \tag{3.8}$$

$$\mathbf{u}(s,t) = 1 \implies \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) \geq p_{\min}^{2^{3^{|Q|^2+1}}}. \tag{3.9}$$

Proof. Consider a finite word $u \in A^*$; by Theorem 3.14 applied to the extended Markov monoid $\mathcal{G}_+$ associated with $\mathcal{A}$ (which is a stabilization monoid thanks to Lemma 3.12) and the morphism $\phi : A \to M$ defined by $\phi(a) = (\mathbf{a}, \mathbf{a})$, there exists a $\sharp$-factorization tree of depth at most $3 \cdot |\mathcal{G}_+|$, whose root is labelled by $(u, (\mathbf{u}, \mathbf{u}_+))$ for some extended limit-word $(\mathbf{u}, \mathbf{u}_+)$.

The depth of a node in this tree is defined in a bottom-up fashion: the leaves have depth zero, and a node has depth one plus the maximum of the depths of its children.

We prove by a bottom-up induction (on h) that for every node $(u, (\mathbf{u}, \mathbf{u}_+))$ of this tree at depth h, for all states s, t:

$$\mathbf{u}_+(s,t) = 1 \iff \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) > 0, \tag{3.10}$$

$$\mathbf{u}(s,t) = 1 \implies \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) \geq p_{\min}^{2^h}. \tag{3.11}$$

The case $h = 0$ is for leaves. Here, either u is a letter a and $\mathbf{u} = \mathbf{u}_+ = \mathbf{a}$, or u is the empty word ε and $\mathbf{u} = \mathbf{u}_+ = \mathbf{1}$. Then both (3.10) and (3.11) hold.

Assume $h > 0$, there are two cases.

First case: a concatenation node labelled by $(u, (\mathbf{u}, \mathbf{u}_+))$ with two children labelled by $(u_1, (\mathbf{u}_1, \mathbf{u}_{+,1}))$ and $(u_2, (\mathbf{u}_2, \mathbf{u}_{+,2}))$. By definition $u = u_1 \cdot u_2$, $\mathbf{u} = \mathbf{u}_1 \cdot \mathbf{u}_2$ and $\mathbf{u}_+ = \mathbf{u}_{+,1} \cdot \mathbf{u}_{+,2}$.

We first prove that (3.10) holds. Indeed, for $s, t \in Q$, $\mathbf{u}_+(s,t) = 1$ if and only if there exists $r \in Q$ such that $\mathbf{u}_{+,1}(s,r) = 1$ and $\mathbf{u}_{+,2}(r,t) = 1$. On the other side, since:

$$\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) = \sum_{r \in Q} \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} r) \cdot \mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_2} t),$$

then $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) > 0$ if and only if there exists $r \in Q$ such that $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} r) \cdot \mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_2} t) > 0$, which is equivalent to $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} r) > 0$ and $\mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_2} t) > 0$. We conclude with the induction hypothesis.

Now we prove that (3.11) holds. Let $s, t \in Q$ such that $\mathbf{u}(s,t) = 1$. Then there exists $r \in Q$ such that $\mathbf{u}_1(s,r) = 1$ and $\mathbf{u}_2(r,t) = 1$. So:

$$\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) \geq \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} r) \cdot \mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_2} t) \geq p_{\min}^{2^h} \cdot p_{\min}^{2^h} = p_{\min}^{2^{h+1}},$$

where the second inequality is by induction hypothesis. This completes the proof of (3.11).

Second case: an iteration node labelled by $(u, (\mathbf{u}^{\sharp}, \mathbf{u}_+))$ with k sons labelled by $(u_1, (\mathbf{u}, \mathbf{u}_+)), \ldots, (u_k, (\mathbf{u}, \mathbf{u}_+))$. By definition, $u = u_1 \cdots u_k$, and $(\mathbf{u}, \mathbf{u}_+)$ is idempotent.

The proof that (3.10) holds is similar to the concatenation node case.

Now we prove that (3.11) holds. Let $s, t \in Q$ such that $\mathbf{u}^\sharp(s, t) = 1$. Since $k \geq 3$:

$$\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) \geq \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} t) \cdot \sum_{q \in Q} \mathbb{P}_{\mathcal{A}}(t \xrightarrow{u_2 \cdots u_{k-1}} q) \cdot \mathbb{P}_{\mathcal{A}}(q \xrightarrow{u_k} t). \tag{3.12}$$

To establish (3.11) we prove that:

$$\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} t) \geq p_{\min}^{2^h}, \tag{3.13}$$

$$\text{for all } q \in Q, \ \mathbb{P}_{\mathcal{A}}(t \xrightarrow{u_2 \cdots u_{k-1}} q) > 0 \implies \mathbb{P}_{\mathcal{A}}(q \xrightarrow{u_k} t) \geq p_{\min}^{2^h}. \tag{3.14}$$

We prove (3.13). Since $\mathbf{u}^\sharp(s, t) = 1$, by definition $\mathbf{u}(s, t) = 1$ and t is $\mathbf{u}$-recurrent. The induction hypothesis for the node $(u_1, (\mathbf{u}, \mathbf{u}_+))$ implies that $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} t) \geq p_{\min}^{2^h}$, *i.e.* (3.13).

Now we prove (3.14). For that we use the hypothesis that $(\mathbf{u}, \mathbf{u}_+)$ is not a leak witness. Let $q \in Q$ such that $\mathbb{P}_{\mathcal{A}}(t \xrightarrow{u_2 \cdots u_{k-1}} q) > 0$. By induction hypothesis for each child, (3.10) implies that $\mathbf{u}_+^{k-2}(t, q) = 1$. Since $\mathbf{u}_+$ is idempotent, $\mathbf{u}_+(t, q) = 1$. We argue that $\mathbf{u}(q, t) = 1$. Let $\ell \in Q$ a $\mathbf{u}$-recurrent state such that $\mathbf{u}(q, \ell) = 1$. Then $\mathbf{u}_+(t, \ell) = 1$, and t, ℓ are $\mathbf{u}$-recurrent. Since $(\mathbf{u}, \mathbf{u}_+)$ is not a leak witness, it follows that $\mathbf{u}(t, \ell) = 1$, which implies that $\mathbf{u}(\ell, t) = 1$ since t is $\mathbf{u}$-recurrent. Together with $\mathbf{u}(q, \ell) = 1$, this implies $\mathbf{u}(q, t) = 1$. Thus, by induction hypothesis and according to (3.11), $\mathbb{P}_{\mathcal{A}}(q \xrightarrow{u_k} t) \geq p_{\min}^{2^h}$, so (3.14) holds.

Now, putting (3.12), (3.13) and (3.14) altogether:

$$\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u} t) \geq \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_1} t) \cdot \sum_{q \in Q} \mathbb{P}_{\mathcal{A}}(t \xrightarrow{u_2 \cdots u_{k-1}} q) \cdot \mathbb{P}_{\mathcal{A}}(q \xrightarrow{u_k} t)$$

$$\geq p_{\min}^{2^h} \cdot \sum_{q \in Q} \mathbb{P}_{\mathcal{A}}(t \xrightarrow{u_2 \cdots u_{k-1}} q) \cdot p_{\min}^{2^h}$$

$$= p_{\min}^{2^{h+1}},$$

where the last equality holds because $\sum_{q \in Q} \mathbb{P}_{\mathcal{A}}(t \xrightarrow{u_2 \cdots u_{k-1}} q) = 1$. This completes the proof of (3.11).

To conclude, note that $\mathcal{G}_+$ has less than $3^{|Q|^2}$ elements. $\qquad\square$

3.6. Completeness of the Markov monoid algorithm for leaktight automata.

In this subsection we rely on the lower bound lemma (Lemma 3.15) to prove Theorem 3.3. Let $\mathcal{A}$ be a leaktight automaton. By Lemma 3.10, its extended Markov monoid does not contain any leak witness, hence Lemma 3.15 applies.

We prove the completeness of the Markov monoid associated with $\mathcal{A}$. Let $(u_n)_{n \in \mathbb{N}}$ be a sequence of finite words. By Lemma 3.15, for each word u_n there exists $(\mathbf{u}_n, \mathbf{u}_{+,n})$ in the extended Markov monoid such that for all states s, t:

$$\mathbf{u}_n(s, t) = 1 \implies \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) \geq p_{\min}^{2^{3^{|Q|^2}+1}}.$$

Since the set of limit-words is finite, there exists $N \in \mathbb{N}$ such that $\{n \in \mathbb{N} \mid \mathbf{u}_N = \mathbf{u}_n\}$ is infinite. To complete the proof, we prove that $\mathbf{u}_N$ satisfies, for all states s, t:

$$\limsup \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) = 0 \implies \mathbf{u}_N(s, t) = 0.$$

Assume $\limsup \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) = 0$, then $\limsup \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) < p_{\min}^{2^{3|Q|^2+1}}$ for n sufficiently large. Since $\mathbf{u}_N = \mathbf{u}_n$ for infinitely many $n \in \mathbb{N}$, this implies $\mathbf{u}_N(s, t) = 0$, which completes the proof of Theorem 3.3.

4. Properties of leaktight automata

In this section, we extend the algorithm presented in Section 2, and investigate its running complexity. The extended algorithm has two features: first, it checks *at the same time* whether an automaton is leaktight and whether it contains a value 1 witness, second, it runs in polynomial space.

We present an algebraic characterization of the leaktight property based on the extended Markov monoid, allowing the extended algorithm to check the leaktight property. For the complexity, one needs a deeper understanding of the Markov monoid; in this section, we will show a linear bound on the $\sharp$-height, allowing to compute the extended Markov monoid in polynomial space. As a corollary, we obtain that the value 1 problem for leaktight automata is PSPACE-complete.

4.1. **Characterization of the leaktight property.** In this subsection, we show the converse of Lemma 3.10, which implies the following theorem, characterizing the leaktight property in algebraic terms.

Theorem 4.1. *An automaton $\mathcal{A}$ is leaktight if and only if its extended Markov monoid does not contain any leak witness.*

Lemma 3.15 is instrumental in the proof of this lemma.

Proof. We prove that if the extended Markov monoid of an automaton $\mathcal{A}$ does not contain any leak witness, then $\mathcal{A}$ is leaktight. The converse was proved in Lemma 3.10.

Assume $\mathcal{A}$ has a leak $(u_n)_{n \in \mathbb{N}}$, we show that its extended Markov monoid contains a leak witness. Consider the Markov chain $\mathcal{M}$ with state space Q and transition matrix M defined by $M(s, t) = \lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t)$. By assumption M is idempotent.

By definition of a leak:

$$r \text{ and } q \text{ are recurrent in } \mathcal{M}, \tag{4.1}$$

$$M(r, q) = 0, \tag{4.2}$$

$$\text{for all } n \in \mathbb{N}, \ \mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_n} q) > 0. \tag{4.3}$$

Assume towards contradiction that the extended Markov monoid does not contain any leak witness, then Lemma 3.15 applies. For each word u_n, there exists $(\mathbf{u}_n, \mathbf{u}_{+,n})$ in the extended Markov monoid such that for all states s, t:

$$\mathbf{u}_{+,n}(s, t) = 1 \iff \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) > 0, \tag{4.4}$$

$$\mathbf{u}_n(s, t) = 1 \implies \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) \geq p_{\min}^{2^{3|Q|^2+1}}. \tag{4.5}$$

Since the extended Markov monoid is finite, there exists $N \in \mathbb{N}$ such that for infinitely many $n \in \mathbb{N}$, we have $(\mathbf{u}_N, \mathbf{u}_{+,N}) = (\mathbf{u}_n, \mathbf{u}_{+,n})$.

Note that since each u_n is idempotent, (4.4) implies that each $\mathbf{u}_{+,n}$ is idempotent as well.

Let $(\mathbf{v}, \mathbf{v}_+) = (\mathbf{u}_N, \mathbf{u}_{+,N})^{|Q|!}$. The power $|Q|!$ ensures that $\mathbf{u}_N^{|Q|!}$ is idempotent, by Lemma 2.2. Since $\mathbf{u}_{+,N}$ is idempotent, $\mathbf{v}_+ = \mathbf{u}_{+,N}$. Also, since $\mathbf{v}$ is idempotent, there exists r' and q' which are $\mathbf{v}$-recurrent, such that $\mathbf{v}(r, r') = 1$ and $\mathbf{v}(q, q') = 1$, again thanks to Lemma 2.2.

Now, we prove that $(\mathbf{v}, \mathbf{v}_+)$ is a leak witness:

$$r' \text{ and } q' \text{ are } \mathbf{v}\text{-recurrent,} \tag{4.6}$$

$$\mathbf{v}(r', q') = 0, \tag{4.7}$$

$$\mathbf{v}_+(r', q') = 1. \tag{4.8}$$

Let $\eta = p_{\min}^{2^{3^{|Q|^2+1}}}$ and $K = |Q|!$.

Observe that for all states s, t, we have $\mathbf{v}(s, t) = 1 \implies M(s, t) > 0$:

$\mathbf{v}(s, t) = 1$

$\implies \mathbf{u}_N^K(s, t) = 1$ (by definition of $\mathbf{v}$)

$\implies \mathbf{u}_n^K(s, t) = 1$ for infinitely many n (by definition of N)

$\implies \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n^K} t) \geq \eta^K$ for infinitely many n (by (4.5))

$\implies \lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n^K} t) \geq \eta^K$

$\implies M^K(s, t) > 0$ (by definition of M)

$\implies M(s, t) > 0$ (since M is idempotent).

First, (4.6) is by definition of r' and q'.

We prove (4.7). Towards contradiction, assume that $\mathbf{v}(r', q') = 1$. Then $M(r', q') > 0$, so together with $M(r, r') > 0$ (which follows from $\mathbf{v}(r, r') = 1$) this implies $M^2(r, q') > 0$, so $M(r, q') > 0$ as M is idempotent. Since $M(q, q') > 0$ (which follows from $\mathbf{v}(q, q') = 1$) and q is recurrent in M, we have $M(q', q) > 0$. This implies $M^2(r, q) > 0$, and $M(r, q) > 0$ because M is idempotent, which contradicts (4.2).

We prove (4.8). Thanks to (4.3) and (4.4), we have $\mathbf{u}_{+,N}(r, q) = 1$, *i.e.* $\mathbf{v}_+(r, q) = 1$. Since $M(r, r') > 0$ and r is recurrent in M, we have $M(r', r) > 0$, so (4.4) implies that $\mathbf{u}_{+,N}(r', r) = 1$, *i.e.* $\mathbf{v}_+(r', r) = 1$. Similarly, $M(q, q') > 0$, so (4.4) implies that $\mathbf{u}_{+,N}(q, q') = 1$, *i.e.* $\mathbf{v}_+(q, q') = 1$. The three equalities $\mathbf{v}_+(r', r) = 1$, $\mathbf{v}_+(r, q) = 1$ and $\mathbf{v}_+(q, q') = 1$ imply $\mathbf{v}_+^3(r', q') = 1$, and since $\mathbf{v}_+$ is idempotent $\mathbf{v}_+(r', q') = 1$.

It follows that $(\mathbf{v}, \mathbf{v}_+)$ is a leak witness, which completes the proof. $\square$

The immediate corollary of Theorem 4.1 is that checking whether an automaton is leaktight can be done by computing the extended Markov monoid and looking for leak witnesses, hence it is decidable.

4.2. The extended Markov monoid algorithm.
Algorithm 2 computes the extended Markov monoid, and looks for value 1 witnesses, which in the extended Markov monoid is an extended limit-word $(\mathbf{u}, \mathbf{u}_+)$ such that $\mathbf{u}$ is a value 1 witness (in the Markov monoid). If there is a value 1 witness, then the automaton has value 1, even if it is not leaktight, thanks to Theorem 2.10. Otherwise, the algorithm looks for a leak witness; if there is no leak witness, then the automaton is leaktight thanks to Theorem 4.1, and it does not have value

1 thanks to Theorem 3.3. In case there is a leak witness, the automaton is not leaktight, and nothing can be said.

ALGORITHM 2: The extended Markov monoid algorithm.

Data: A probabilistic automaton.
$\mathcal{G}_+ \leftarrow \{(\mathbf{a}, \mathbf{a}) \mid a \in A\} \cup \{(\mathbf{1}, \mathbf{1})\}$.
repeat
  if *there is* $(\mathbf{u}, \mathbf{u}_+), (\mathbf{v}, \mathbf{v}_+) \in \mathcal{G}_+$ *such that* $(\mathbf{u} \cdot \mathbf{v}, \mathbf{u}_+ \cdot \mathbf{v}_+) \notin \mathcal{G}_+$ **then**
    | add $(\mathbf{u} \cdot \mathbf{v}, \mathbf{u}_+ \cdot \mathbf{v}_+)$ to $\mathcal{G}_+$
  end
  if *there is* $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+$ *such that* $(\mathbf{u}, \mathbf{u}_+)$ *is idempotent and* $(\mathbf{u}^\sharp, \mathbf{u}_+) \notin \mathcal{G}_+$ **then**
    | add $(\mathbf{u}^\sharp, \mathbf{u}_+)$ to $\mathcal{G}_+$
  end
until *there is nothing to add*;
if *there is a value* 1 *witness in* $\mathcal{G}_+$ **then**
  | return true;
else
  if *there is no leak witness in* $\mathcal{G}_+$ **then**
    | return false;
  else
    | return fail: the automaton is not leaktight;
  end
end

4.3. **Parallel composition and** PSPACE**-hardness.** The objective of this subsection is to prove the PSPACE-hardness of the value 1 problem for leaktight automata. To this end, we give a reduction from the emptiness problem of n deterministic automata. To prove that the reduction indeed constructs leaktight automata, we need to show that deterministic automata are leaktight, and the closure under parallel composition.

Proposition 4.2. *Deterministic automata are leaktight.*

Proof. For all limit-words $\mathbf{u} \in \{\mathbf{a} \mid a \in A\} \cup \{\mathbf{1}\}$, for all states s, there exists a unique state t such that $\mathbf{u}(s, t) = 1$. In particular, each recurrence class is formed of only one state with a self-loop. This property is preserved by concatenation, and implies that the iteration operation is trivial, *i.e.* $\mathbf{u}^\sharp = \mathbf{u}$. Consequently, for all extended limit-words $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid, we have $\mathbf{u} = \mathbf{u}_+$, which implies that there are no leak witnesses. $\square$

Definition 4.3 (Parallel composition)**.** *Consider two probabilistic automata, denoted* $\mathcal{A} = (Q^\mathcal{A}, q_0^\mathcal{A}, \Delta^\mathcal{A}, F^\mathcal{A})$ *and* $\mathcal{B} = (Q^\mathcal{B}, q_0^\mathcal{B}, \Delta^\mathcal{B}, F^\mathcal{B})$. *We assume that* $Q^\mathcal{A}$ *and* $Q^\mathcal{B}$ *are disjoint.*
  The parallel composition of $\mathcal{A}$ *and* $\mathcal{B}$ *is:*

$$\mathcal{A} \parallel \mathcal{B} = (Q^\mathcal{A} \uplus Q^\mathcal{B}, \ \delta_0, \ \Delta, \ F^\mathcal{A} \cup F^\mathcal{B}),$$

where $\delta_0 = \frac{1}{2} \cdot q_0^\mathcal{A} + \frac{1}{2} \cdot q_0^\mathcal{B}$, *and:*

$$\Delta(q, a) = \begin{cases} \Delta_\mathcal{A}(q, a) & \text{if } q \in Q_\mathcal{A}, \\ \Delta_\mathcal{B}(q, a) & \text{if } q \in Q_\mathcal{B}. \end{cases}$$

By definition, for $u \in A^*$, we have $\mathbb{P}_{\mathcal{A}||\mathcal{B}}(u) = \frac{1}{2} \cdot \mathbb{P}_{\mathcal{A}}(u) + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{B}}(u)$. Note that in this definition, we allowed an initial probability distribution rather than only one initial state. This could be avoided by adding a new initial state that leads to each previous initial state with probability half, but we do it here for technical convenience in the proof of the following proposition.

Proposition 4.4. *The leaktight property is stable by parallel composition.*

Proof. The extended Markov monoid $\mathcal{G}_+^{\mathcal{A}||\mathcal{B}}$ of the parallel composition embeds into the direct product $\mathcal{G}_+^{\mathcal{A}} \times \mathcal{G}_+^{\mathcal{B}}$ of the extended Markov monoids of each automaton.

Note that for $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+^{\mathcal{A}||\mathcal{B}}$, if $\mathbf{u}(s,t) = 1$, then either $s, t \in Q^{\mathcal{A}}$ or $s, t \in Q^{\mathcal{B}}$, and similarly for $\mathbf{u}_+$. Relying on this, we map $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+^{\mathcal{A}||\mathcal{B}}$ to $((\mathbf{u}, \mathbf{u}_+)[\mathcal{A}], (\mathbf{u}, \mathbf{u}_+)[\mathcal{B}])$, where $(\mathbf{u}, \mathbf{u}_+)[\mathcal{A}]$ is the restriction to $\mathcal{A}$ and similarly for $\mathcal{B}$. An easy induction on $(\mathbf{u}, \mathbf{u}_+)$ shows that this map is an embedding into $\mathcal{G}_+^{\mathcal{A}} \times \mathcal{G}_+^{\mathcal{B}}$.

Consequently, the extended Markov monoid of the parallel composition contains a leak witness if and only if one of the extended Markov monoid contains a leak witness. □

Now that we proved that deterministic automata are leaktight, and the closure under parallel composition, the PSPACE-hardness of the value 1 problem for leaktight automata is easy.

Proposition 4.5. *The value 1 problem for leaktight automaton is* PSPACE-*hard.*

Proof. We give a reduction from the following problem: given n deterministic automata over finite words, decide whether the intersection of the languages they accept is empty. This problem is PSPACE-hard [Koz77].

The reduction is as follows: given n deterministic automata, we construct the parallel composition of the n automata, where each copy is reached with probability $\frac{1}{n}$. This automaton has value 1 if and only if the intersection of the languages is not empty, and is leaktight by Proposition 4.2 and Proposition 4.4. □

4.4. **Bounding the $\sharp$-height in the Markov monoid.** We now consider the running complexity of the extended Markov monoid algorithm. A naïve argument shows that it terminates in less than $3^{|Q|^2}$ iterations, since each iteration adds a new extended limit-word in the monoid and there are at most $3^{|Q|^2}$ different limit-words. This gives an EXPTIME upper bound.

A better complexity can be achieved by looking for a value 1 witness or a leak witness in a non-deterministic way. The algorithm guesses the witness by its decomposition into concatenations and iterations. The key observation, made by Kirsten [Kir05] in the context of distance desert automata, is that the $\sharp$-height, that is the number of nested applications of the iteration operation, can be restricted to at most $|Q|$.

Note that when dealing with $\sharp$-height, it suffices to consider limit-words instead of extended limit-words, as by definition the second component of an extended limit-word does not contain any $\sharp$.

Formally, we define the $\sharp$-hierarchy inside the Markov monoid as follows:

$$S_0 = \langle \{\mathbf{a} \mid a \in A\} \cup \{\mathbf{1}\} \rangle,$$
$$S_{p+1} = \langle S_p \cup \{\mathbf{u}^\sharp \mid \mathbf{u} \in E(S_p)\} \rangle,$$

where $\langle T \rangle$ is the set of limit-words obtained as concatenation of limit-words in T.

Definition 4.6 ($\sharp$-height of a limit-word)**.** *The $\sharp$-height of a limit-word $\mathbf{u}$ is the minimal p such that $\mathbf{u} \in S_p$.*

Theorem 4.7. *Every limit-word has $\sharp$-height at most $|Q|$, i.e. the $\sharp$-hierarchy collapses at level $|Q|$.*

In the following, we adapt Kirsten's proof from [Kir05] to the setting of probabilistic automata. Roughly speaking, the proof consists in associating a quantity to each idempotent element of the Markov monoid, and to show the following:

- the quantity is bounded above by $|Q|$.
- the quantity strictly decreases when iterating an unstable limit-word (*i.e.* if $\mathbf{u}^\sharp \neq \mathbf{u}$),
- the quantity does not increase when concatenating.

Let $\mathbf{u}$ be an idempotent limit-word, we define $\sim_\mathbf{u}$ the relation on Q by $s \sim_\mathbf{u} t$ if $\mathbf{u}(s,t) = 1$ and $\mathbf{u}(t,s) = 1$. Clearly, $\sim_\mathbf{u}$ is symmetric, and since $\mathbf{u}$ is idempotent, $\sim_\mathbf{u}$ is transitive. If for some state s there exists a state t such that $s \sim_\mathbf{u} t$, then $s \sim_\mathbf{u} s$ since $\mathbf{u}$ is idempotent. Consequently, the restriction of $\sim_\mathbf{u}$ to the set

$$Z_\mathbf{u} = \{s \in Q \mid s \sim_\mathbf{u} s\}$$

is reflexive, *i.e.* $\sim_\mathbf{u}$ is an equivalence relation on $Z_\mathbf{u}$. From now on by equivalence class of $\sim_\mathbf{u}$ we mean an equivalence class of $\sim_\mathbf{u}$ on $Z_\mathbf{u}$. We denote by $[s]_\mathbf{u}$ the equivalence class of s, and by $\mathbf{Cl}(\mathbf{u})$ the set of equivalence classes of $\sim_\mathbf{u}$. The quantity associated with $\mathbf{u}$ is $|\mathbf{Cl}(\mathbf{u})|$, the number of equivalence classes of $\sim_\mathbf{u}$, that is the number of non-trivial connected components in the underlying graph of $\mathbf{u}$. Note that $|\mathbf{Cl}(\mathbf{u})| \leq |Q|$.

Here are two useful observations.

Lemma 4.8.
- *Let $\mathbf{u}, \mathbf{v}$ be two limit-words and $s,t,r \in Q$. Then $(\mathbf{u} \cdot \mathbf{v})(s,t) \geq \mathbf{u}(s,r) \cdot \mathbf{v}(r,t)$.*
- *Let $\mathbf{u}$ be an idempotent limit-word and $s,t \in Q$. There exists $r \in Q$ such that $\mathbf{u}(s,t) = \mathbf{u}(s,r) \cdot \mathbf{u}(r,r) \cdot \mathbf{u}(r,t)$.*

Proof. The first claim is clear and follows from the equality:

$$(\mathbf{u} \cdot \mathbf{v})(s,t) = \sum_{r \in Q} \mathbf{u}(s,r) \cdot \mathbf{v}(r,t).$$

Consider now the second claim. For all states $r \in Q$, since $\mathbf{u}$ is idempotent we have:

$$\mathbf{u}(s,t) = \mathbf{u}^3(s,t) = \sum_{p,q \in Q} \mathbf{u}(s,p) \cdot \mathbf{u}(p,q) \cdot \mathbf{u}(q,t) \geq \mathbf{u}(s,r) \cdot \mathbf{u}(r,r) \cdot \mathbf{u}(r,t).$$

Since $\mathbf{u}$ is idempotent, we have $\mathbf{u} = \mathbf{u}^{n+2}$, so there exist $s = r_0, \ldots, r_{n+2} = t$ such that $\mathbf{u}(s,t) = \mathbf{u}(r_0, r_1) \cdots \mathbf{u}(r_{n+1}, r_{n+2})$. By a counting argument, there exist i,j such that $1 \leq i < j \leq (n+1)$ and $r_i = r_j$, denote it by r. We have:

$$\mathbf{u}(s,r) = \mathbf{u}^i(s,r) \geq \mathbf{u}(r_0, r_1) \cdots \mathbf{u}(r_{i-1}, r_i),$$

$$\mathbf{u}(r,r) = \mathbf{u}^{j-i}(r,r) \geq \mathbf{u}(r_i, r_{i+1}) \cdots \mathbf{u}(r_{j-1}, r_j),$$

$$\mathbf{u}(r,t) = \mathbf{u}^{n+2-j}(r,t) \geq \mathbf{u}(r_j, r_{j+1}) \cdots \mathbf{u}(r_{n+1}, r_{n+2}).$$

Hence, $\mathbf{u}(s,r) \cdot \mathbf{u}(r,r) \cdot \mathbf{u}(r,t) \geq \mathbf{u}(r_0, r_1) \cdots \mathbf{u}(r_{n+1}, r_{n+2}) = \mathbf{u}(s,t)$, and the second claim follows. $\qquad\square$

The following lemma shows that the quantity $|\mathbf{Cl}(\mathbf{u})|$ strictly decreases when iterating an unstable limit-word (*i.e.* if $\mathbf{u}^\sharp \neq \mathbf{u}$).

Lemma 4.9. *Let $\mathbf{u}$ be an idempotent limit-word.*

$$\mathbf{Cl}(\mathbf{u}^\sharp) \subseteq \mathbf{Cl}(\mathbf{u}), \tag{4.9}$$

$$\text{if } \mathbf{u} \neq \mathbf{u}^\sharp, \text{ then } \mathbf{Cl}(\mathbf{u}^\sharp) \neq \mathbf{Cl}(\mathbf{u}). \tag{4.10}$$

Proof. We prove (4.9). Let $s \in Z_{\mathbf{u}^\sharp}$; by definition we have $s \sim_{\mathbf{u}^\sharp} s$. We show that $[s]_{\mathbf{u}^\sharp} = [s]_{\mathbf{u}}$. For all states $t \in Q$ such that $s \sim_{\mathbf{u}^\sharp} t$, we have $s \sim_{\mathbf{u}} t$, so $[s]_{\mathbf{u}^\sharp} \subseteq [s]_{\mathbf{u}}$. Conversely, let $t \in [s]_{\mathbf{u}}$; we have $\mathbf{u}(s,t) = 1$ and $\mathbf{u}(t,s) = 1$. Since $s \sim_{\mathbf{u}^\sharp} s$, we have $\mathbf{u}^\sharp(s,s) = 1$. So $\mathbf{u}^\sharp(s,t) = (\mathbf{u}^\sharp \cdot \mathbf{u})(s,t) \geq \mathbf{u}^\sharp(s,s) \cdot \mathbf{u}(s,t) = 1$, and similarly $\mathbf{u}^\sharp(t,s) = (\mathbf{u} \cdot \mathbf{u}^\sharp)(t,s) \geq \mathbf{u}(t,s) \cdot \mathbf{u}^\sharp(s,s) = 1$. Thus $s \sim_{\mathbf{u}^\sharp} t$, *i.e.* $t \in [s]_{\mathbf{u}^\sharp}$, which concludes to the equality $[s]_{\mathbf{u}^\sharp} = [s]_{\mathbf{u}}$. In other words, the equivalence classes for $\mathbf{u}^\sharp$ are also equivalence classes for $\mathbf{u}$, so $\mathbf{Cl}(\mathbf{u}^\sharp) \subseteq \mathbf{Cl}(\mathbf{u})$.

We prove (4.10). Assume $\mathbf{u} \neq \mathbf{u}^\sharp$; let s, t such that $\mathbf{u}(s,t) = 1$ and $\mathbf{u}^\sharp(s,t) = 0$. By Lemma 4.8, there exists r such that $\mathbf{u}(s,t) = \mathbf{u}(s,r) \cdot \mathbf{u}(r,r) \cdot \mathbf{u}(r,t)$, so $\mathbf{u}(r,r) = 1$. Towards contradiction, assume $\mathbf{u}^\sharp(r,r) = 1$. It follows that:

$$\mathbf{u}^\sharp(s,t) = (\mathbf{u} \cdot \mathbf{u}^\sharp \cdot \mathbf{u})(s,t) \geq \mathbf{u}(s,r) \cdot \mathbf{u}^\sharp(r,r) \cdot \mathbf{u}(r,t) = \mathbf{u}(s,r) \cdot \mathbf{u}(r,r) \cdot \mathbf{u}(r,t) = 1,$$

i.e., $\mathbf{u}^\sharp(s,t) = 1$ which is a contradiction. Consequently, $\mathbf{u}^\sharp(r,r) = 0$, so $r \sim_{\mathbf{u}} r$ and $r \not\sim_{\mathbf{u}^\sharp} r$. Thus, $r \in Z_{\mathbf{u}}$ but $r \notin Z_{\mathbf{u}^\sharp}$. Hence, there is a class $[r]_{\mathbf{u}}$ in $\mathbf{Cl}(\mathbf{u})$, but there is no class $[r]_{\mathbf{u}^\sharp}$ in $\mathbf{Cl}(\mathbf{u}^\sharp)$. $\qquad\square$

The following lemma shows that the quantity $|\mathbf{Cl}(\mathbf{u})|$ is common to all idempotents in the same $\mathcal{J}$-class. The notion of $\mathcal{J}$-class is a classical notion for the theory of monoids, derived from one of the four Green's relations called the $\mathcal{J}$-preorder (for details about Green's relations, see [Lal79, How95], or [Col11] for its applications to automata theory).

Define $\mathbf{u} \leq_{\mathcal{J}} \mathbf{v}$ if there exist $\mathbf{a}, \mathbf{b}$ such that $\mathbf{a} \cdot \mathbf{v} \cdot \mathbf{b} = \mathbf{u}$, and $\mathbf{u}\mathcal{J}\mathbf{v}$, *i.e.* $\mathbf{u}$ and $\mathbf{v}$ are in the same $\mathcal{J}$-class, if $\mathbf{u} \leq_{\mathcal{J}} \mathbf{v}$ and $\mathbf{v} \leq_{\mathcal{J}} \mathbf{u}$.

Lemma 4.10. *Let $\mathbf{u}, \mathbf{v}$ be idempotent limit-words. If $\mathbf{u} \leq_{\mathcal{J}} \mathbf{v}$, then $|\mathbf{Cl}(\mathbf{u})| \leq |\mathbf{Cl}(\mathbf{v})|$.*

Proof. Let $\mathbf{a}, \mathbf{b}$ two limit-words such that $\mathbf{a} \cdot \mathbf{v} \cdot \mathbf{b} = \mathbf{u}$. First, without loss of generality we assume that $\mathbf{a} \cdot \mathbf{v} = \mathbf{a}$ and $\mathbf{v} \cdot \mathbf{b} = \mathbf{b}$. Indeed, if $\mathbf{a}$ and $\mathbf{b}$ do not satisfy these conditions, then we consider $\mathbf{a} = \mathbf{a} \cdot \mathbf{v}$ and $\mathbf{b} = \mathbf{v} \cdot \mathbf{b}$.

We construct a partial surjective mapping $\beta : \mathbf{Cl}(\mathbf{v}) \to \mathbf{Cl}(\mathbf{u})$, which depends on the choice of $\mathbf{a}$ and $\mathbf{b}$. For all states $s \in Z_{\mathbf{v}}$ and $t \in Z_{\mathbf{u}}$ satisfying $\mathbf{a}(t,s) \cdot \mathbf{v}(s,s) \cdot \mathbf{b}(s,t) = 1$ we set $\beta([s]_{\mathbf{v}}) = [t]_{\mathbf{u}}$. To complete the proof, we have to show that β is well defined and that β is indeed surjective.

We show that β is well defined. Let $s, s' \in Z_{\mathbf{v}}$ and $t, t' \in Z_{\mathbf{u}}$, and assume $\mathbf{a}(t,s) \cdot \mathbf{v}(s,s) \cdot \mathbf{b}(s,t) = 1$ and $\mathbf{a}(t',s') \cdot \mathbf{v}(s',s') \cdot \mathbf{b}(s',t') = 1$. By definition $\beta([s]_{\mathbf{v}}) = [t]_{\mathbf{u}}$ and $\beta([s']_{\mathbf{v}}) = [t']_{\mathbf{u}}$. To show that β is well defined, we have to show that if $[s]_{\mathbf{v}} = [s']_{\mathbf{v}}$, then $[t]_{\mathbf{u}} = [t']_{\mathbf{u}}$. Assume $[s]_{\mathbf{v}} = [s']_{\mathbf{v}}$, *i.e.*, $s \sim_{\mathbf{v}} s'$, so $\mathbf{v}(s,s') = 1$. Since $\mathbf{a}(t,s) \cdot \mathbf{v}(s,s) \cdot \mathbf{b}(s,t) = 1$, we have $\mathbf{a}(t,s) = \mathbf{b}(s,t) = 1$. Similarly, $\mathbf{a}(t',s') \cdot \mathbf{v}(s',s') \cdot \mathbf{b}(s',t') = 1$, so $\mathbf{a}(t',s') = \mathbf{b}(s',t') = 1$. Consequently, $\mathbf{a}(t,s) \cdot \mathbf{v}(s,s') \cdot \mathbf{b}(s',t') = 1$, so $\mathbf{u}(t,t') = (\mathbf{a} \cdot \mathbf{v} \cdot \mathbf{b})(t,t') = 1$. Symmetrically, $\mathbf{a}(t',s') \cdot \mathbf{v}(s',s) \cdot \mathbf{b}(s,t) = 1$, so $\mathbf{u}(t',t) = 1$, concluding to $t \sim_{\mathbf{u}} t'$, *i.e.* $[t]_{\mathbf{u}} = [t']_{\mathbf{u}}$.

We show that β is surjective. Let $t \in Z_{\mathbf{u}}$. We exhibit some s such that $\beta([s]_{\mathbf{v}}) = [t]_{\mathbf{u}}$. Since $\mathbf{u} = \mathbf{a} \cdot \mathbf{v} \cdot \mathbf{b}$, there are p, q such that $\mathbf{a}(t,p) \cdot \mathbf{v}(p,q) \cdot \mathbf{b}(q,t) = \mathbf{u}(t,t) = 1$, so $\mathbf{a}(t,p) = \mathbf{v}(p,q) = \mathbf{b}(q,t) = 1$. By Lemma 4.8 there exists s such that $\mathbf{v}(p,s) \cdot \mathbf{v}(s,s) \cdot \mathbf{v}(s,q) = \mathbf{v}(p,q) = 1$, so, $\mathbf{v}(p,s) = \mathbf{v}(s,s) = \mathbf{v}(s,q) = 1$. We have $\mathbf{a}(t,s) = (\mathbf{a} \cdot \mathbf{v})(t,s) \geq$

$\mathbf{a}(t, p) \cdot \mathbf{v}(p, s) = 1$, and $\mathbf{b}(s, t) = (\mathbf{v} \cdot \mathbf{b})(s, t) \geq \mathbf{v}(s, q) \cdot \mathbf{b}(q, t) = 1$. To sum up, $\mathbf{a}(t, s) \cdot \mathbf{v}(s, s) \cdot \mathbf{b}(s, t) = 1$, and hence, $\beta([s]_{\mathbf{v}}) = [t]_{\mathbf{u}}$. $\qquad \square$

The following lemma wraps up the previous two lemma. For technical convenience, we set $S_{-1} = \emptyset$.

Lemma 4.11. *Let $\mathbf{u}$ be an idempotent limit-word and $p \geq 0$. If $\mathbf{u} \in S_p \setminus S_{p-1}$, then $|\mathbf{Cl}(\mathbf{u})| \leq |Q| - p$.*

Proof. We proceed by induction on p. For $p = 0$, the assertion is obvious. Let $p \geq 0$, we show the claim for $p + 1$. Let $\mathbf{u}$ be an idempotent limit-word such that $\mathbf{u} \in S_{p+1} \setminus S_p$. By definition, $\mathbf{u} = \mathbf{v}_1 \cdots \mathbf{v}_k$ where for all i, either $\mathbf{v}_i \in S_p$ or $\mathbf{v}_i = \mathbf{u}_i^{\sharp}$ for $\mathbf{u}_i \in S_p$ and $\mathbf{v}_i \notin S_p$.

If for all i, $\mathbf{v}_i \in S_p$, then $\mathbf{u} = \mathbf{v}_1 \cdots \mathbf{v}_k \in S_p$, which is a contradiction. Consequently, there exists i such that $\mathbf{v}_i = \mathbf{u}_i^{\sharp}$ for $\mathbf{u}_i \in S_p$ and $\mathbf{v}_i \notin S_p$. Since $\mathbf{u}_i \in S_p$ and $\mathbf{v}_i = \mathbf{u}_i^{\sharp} \notin S_p$, we have $\mathbf{u}_i^{\sharp} \neq \mathbf{u}_i$. Towards contradiction, assume $\mathbf{u}_i \in S_{p-1}$, then $p \geq 1$, and this implies $\mathbf{u}_i^{\sharp} \in S_p$, which is a contradiction. Hence, $\mathbf{u}_i \in S_p \setminus S_{p-1}$.

By induction, we have $|\mathbf{Cl}(\mathbf{u}_i)| \leq |Q| - p$. Since $\mathbf{u}_i^{\sharp} \neq \mathbf{u}_i$, by Lemma 4.9 we have $|\mathbf{Cl}(\mathbf{u}_i^{\sharp})| < |\mathbf{Cl}(\mathbf{u}_i)|$. Since $\mathbf{u} \leq_{\mathcal{J}} \mathbf{u}_i^{\sharp}$, by Lemma 4.10 we have $|\mathbf{Cl}(\mathbf{u})| \leq |\mathbf{Cl}(\mathbf{u}_i^{\sharp})|$. Altogether, it follows $|\mathbf{Cl}(\mathbf{u})| \leq |Q| - (p + 1)$. $\qquad \square$

It follows from Lemma 4.11 that $S_{|Q|} = S_{|Q|+1}$, *i.e.* the $\sharp$-hierarchy collapses at level $|Q|$, proving Theorem 4.7.

The bound is almost tight, as shown in figure 3. The only value 1 witness of this automaton is $(\cdots((\mathbf{a}_0^{\sharp} \ \mathbf{a}_1)^{\sharp} \ \mathbf{a}_2)^{\sharp} \ \mathbf{a}_3)^{\sharp} \ \cdots \mathbf{a}_{n-1})^{\sharp}$, whose $\sharp$-height is $|Q| - 2$. Note that this
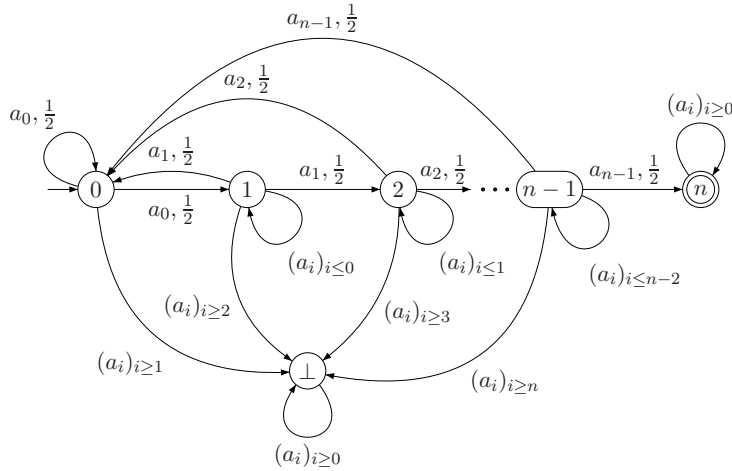

Figure 3: A leaktight automaton with value 1 and $\sharp$-height $|Q| - 2$.

automaton is leaktight, so the extended Markov monoid algorithm will find the value 1 witness and correctly answers that it has value 1.

4.5. **Finding witnesses in the Markov monoid.** In the subsection, we will prove the following complexity result.

Proposition 4.12. *There exists an algorithm which checks in polynomial space whether an automaton is leaktight and whether in such case it has value* 1.

Following Theorem 3.3, checking whether a leaktight automaton has value 1 boils down to finding a value 1 witness in the Markov monoid. Similarly, following Theorem 4.1, checking whether an automaton is *not* leaktight boils down to finding a leak witness in the extended Markov monoid. Note that in both cases, checking whether a given limit-word or extended limit-word is a witness is easily done in polynomial time.

Since we aim at proving that those two tasks can be computed in PSPACE, which is closed under complementation, it suffices to show how to find a witness in the (extended) Markov monoid. For the sake of readability, we here only deal with the Markov monoid, but similar ideas apply to the extended Markov monoid.

We describe an algorithm to guess a witness in the Markov monoid. The key property given by Theorem 4.7 is that we can restrict ourselves to at most $|Q|$ nested iteration operations.

As the corresponding property was proved by Kirsten [Kir05] in the context of distance automata, also to obtain a PSPACE algorithm, the following algorithm is also an adaptation of [Kir05]. Rather than a formal proof, we here give an intuitive description of the algorithm.

A witness can be described as a tree whose nodes are labelled by limit-words, of depth at most $2 \cdot |Q| + 1$, as follows:

- a leaf is labelled either by **a** for $a \in A$ or by **1**,
- an internal node can be a *concatenation node*, then it is labelled by $\mathbf{u} = \mathbf{v}_1 \cdots \mathbf{v}_k$ for $k \leq 2^{|Q|^2}$ and has k children, labelled by $\mathbf{v}_1, \ldots, \mathbf{v}_k$,
- an internal node can be an *iteration node*, then it is labelled by $\mathbf{u}^\sharp$ and has one child labelled **u**.

We describe an algorithm that guesses such a tree. It starts from the root, and travels over nodes in a depth-first way: from top to bottom (and up again) and from left to right. In a node, the algorithm stores the branch that leads to this node, and for each node in the branch the limit-word obtained by concatenating all the left siblings of this node. From a node, the algorithm guesses a limit-word, and whether it will be a leaf, a concatenation node or an iteration node. In the first case, it goes up and checks the consistency of this guess. In the two other cases, it updates the value of this node by concatenating the new guess with the previous value and goes down.

Although the tree is of exponential size, in each step the algorithm only stores $2 \cdot |Q| + 1$ limit-words at most, so it runs in polynomial space.

5. Examples and subclasses of leaktight automata

In this section, we investigate further the class of leaktight automata, by giving examples of leaktight automata, exhibiting subclasses, and showing closure properties. In particular, we prove that hierarchical automata, $\sharp$-acylic automata and simple automata are all strict subclasses of leaktight automata. (Actually, since $\sharp$-acylic automata are already a subclass of simple automata, we do not consider them.) This implies that our decidability result extends the decidability results from [GO10, CT12].

5.1. **Two basic examples.** The automaton on figure 4 is leaktight. As we shall see, it is not hierarchical, nor simple, hence it witnesses that leaktight automata are not subsumed by hierarchical or simple automata. Its extended Markov monoid is depicted on the right-hand side. Each of the four directed graphs represents an extended limit-word $(\mathbf{u}, \mathbf{u}_+)$: if $\mathbf{u}(s,t) = 1$, then (s,t) is an edge, and if $\mathbf{u}(s,t) = 0$ but $\mathbf{u}_+(s,t) = 1$, then (s,t) is marked with $+$.

The initial state of the automaton is state 0, and the unique final state is state 1. This automaton has value 1 and this can be checked using the extended Markov monoid: the two value 1 witnesses are $\mathbf{a}^\sharp$ and $\mathbf{b} \cdot \mathbf{a}^\sharp$.
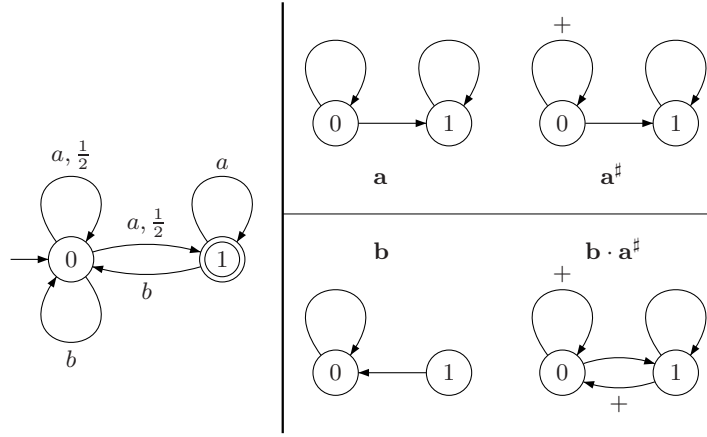

Figure 4: A leaktight automaton and its extended Markov monoid.

The automaton on figure 5 is leaktight. The initial state of the automaton is state 0, and the unique final state is state F. The Markov monoid has too many elements to be represented here. This automaton does not have value 1.
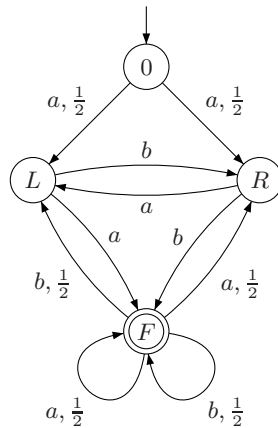

Figure 5: A leaktight automaton which does not have value 1.

5.2. **Some closure properties.** In this subsection, we show some closure properties: synchronised product and composition with a deterministic transducer. As for Proposition 4.4, the automata constructions reflect in algebraic constructions, allowing to use the characterization with leak witnesses given by Theorem 4.1.

In this subsection, we will omit initial and final states, as we are only interested in preserving the leaktight property, which does not depend on it.

Definition 5.1 (Synchronised product). *Let $\mathcal{A} = (Q^{\mathcal{A}}, \Delta^{\mathcal{A}})$ and $\mathcal{B} = (Q^{\mathcal{B}}, \Delta^{\mathcal{B}})$ be two probabilistic automata.*

A synchronised product of $\mathcal{A}$ and $\mathcal{B}$ is:

$$\mathcal{A} \times \mathcal{B} \; = \; (Q^{\mathcal{A}} \times Q^{\mathcal{B}} \; , \; \Delta),$$

where $\Delta(q, a) = (\Delta_{\mathcal{A}}(q, a), \Delta_{\mathcal{B}}(q, a))$.

Proposition 5.2. *The leaktight property is stable by synchronized product.*

Proof. The extended Markov monoid $\mathcal{G}_+^{\mathcal{A} \times \mathcal{B}}$ of the synchronized product embeds into the direct product $\mathcal{G}_+^{\mathcal{A}} \times \mathcal{G}_+^{\mathcal{B}}$ of the extended Markov monoids of each automaton.

Let $(\mathbf{u}, \mathbf{u}_+)$ be an extended limit-word in $\mathcal{G}_+^{\mathcal{A} \times \mathcal{B}}$. Define $\mathbf{u}_{\mathcal{A}}(s, t) = 1$ if there exists $s', t' \in Q^{\mathcal{B}}$ such that $\mathbf{u}((s, s'), (t, t')) = 1$, and similarly for $\mathbf{u}_{+,\mathcal{A}}$, $\mathbf{u}_{\mathcal{B}}$ and $\mathbf{u}_{+,\mathcal{B}}$. We have the following equivalence:

$$\mathbf{u}((s, s'), (t, t')) = 1 \iff \mathbf{u}_{\mathcal{A}}(s, t) = 1 \wedge \mathbf{u}_{\mathcal{B}}(s', t') = 1,$$

and similarly for $\mathbf{u}_+$.

Relying on this, we map $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+^{\mathcal{A} \times \mathcal{B}}$ to $((\mathbf{u}_{\mathcal{A}}, \mathbf{u}_{+,\mathcal{A}}) \; , \; (\mathbf{u}_{\mathcal{B}}, \mathbf{u}_{+,\mathcal{B}}))$. An easy induction shows that this map is an embedding into $\mathcal{G}_+^{\mathcal{A}} \times \mathcal{G}_+^{\mathcal{B}}$.

Consequently, the extended Markov monoid of the synchronised product contains a leak witness if and only if one of the extended Markov monoid contains a leak witness. $\square$

The last closure property we prove will be useful in the next section.

Definition 5.3 (Composition with a deterministic transducer). *Let $\mathcal{A} = (Q^{\mathcal{A}}, \Delta^{\mathcal{A}})$ be a probabilistic automaton, and $\mathcal{M} = (Q^{\mathcal{M}}, Q^{\mathcal{A}}, \Delta^{\mathcal{M}})$ a deterministic transducer over $\mathcal{A}$, i.e. $\Delta^{\mathcal{M}} : Q^{\mathcal{M}} \times Q^{\mathcal{A}} \to Q^{\mathcal{M}}$.*

The composition of $\mathcal{A}$ by $\mathcal{M}$ is:

$$\mathcal{A} \otimes \mathcal{M} \; = \; (Q^{\mathcal{A}} \times Q^{\mathcal{M}} \; , \; \Delta),$$

where $\Delta(q, p, a) = (\Delta_{\mathcal{A}}(q, a), \Delta_{\mathcal{M}}(p, q))$.

Proposition 5.4. *The leaktight property is stable by composition with a deterministic transducer.*

Proof. Following the same reasoning as in 4.2, one can show that the extended Markov monoids for $\mathcal{A}$ and for $\mathcal{A} \otimes \mathcal{M}$ are isomorphic. $\square$

5.3. **Leaktight automata strictly contain hierarchical automata.** The class of hierarchical automata has been defined in [CSV11], where it was proved that they are recognize exactly the class of ω-regular languages. The states Q of a hierarchical automaton are sorted according to levels such that for each letter, at most one successor is at the same level and all others are at higher levels.

Formally, there exists a mapping rank $: Q \to \mathbb{N}$ such that for all $a \in A$, for all states s, t such that $\mathbb{P}_{\mathcal{A}}(s \overset{a}{\to} t) > 0$, we have rank$(s) \leq$ rank(t). Furthermore, if $\mathbb{P}_{\mathcal{A}}(s \overset{a}{\to} t) > 0$ and $\mathbb{P}_{\mathcal{A}}(s \overset{a}{\to} t') > 0$ but rank$(s) =$ rank$(t) =$ rank(t'), then $t = t'$.

Proposition 5.5. *Every hierarchical automata is leaktight.*

Proof. We prove by induction that for every extended limit-word $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid of a hierarchical automaton, for every states s, t, t':

$$\mathbf{u}_+(s,t) = 1 \implies \text{rank}(s) \leq \text{rank}(t), \qquad (5.1)$$

$$\mathbf{u}_+(s,t) = 1 \ \wedge \ \mathbf{u}_+(s,t') = 1 \ \wedge \ \text{rank}(s) = \text{rank}(t) = \text{rank}(t') \implies t = t', \qquad (5.2)$$

$$\mathbf{u}(s,t) = 1 \implies \text{rank}(s) \leq \text{rank}(t), \qquad (5.3)$$

$$\mathbf{u}(s,t) = 1 \ \wedge \ \mathbf{u}(s,t') = 1 \ \wedge \ \text{rank}(s) = \text{rank}(t) = \text{rank}(t') \implies t = t'. \qquad (5.4)$$

Note that (5.1) and (5.2) imply (5.3) and (5.4), since $\mathbf{u}(s,t) = 1$ implies $\mathbf{u}_+(s,t) = 1$. The key property following from (5.2) and (5.3) is that the recurrence classes of $\mathbf{u}$ and of $\mathbf{u}_+$ are singletons.

This is trivial for $(\mathbf{1}, \mathbf{1})$. The case of $(\mathbf{a}, \mathbf{a})$ is the definition of hierarchical automata. The induction step for concatenation is routinely checked, and trivial for the iteration.

We now prove that the extended Markov monoid of a hierarchical automaton does not contain any leak witness. We prove a slightly stronger statement; for every extended limit-word $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid of a hierarchical automaton, for all states q, r:

$$\mathbf{u}_+(r,q) = 1 \ \wedge \ r \text{ is } \mathbf{u}\text{-recurrent} \implies \mathbf{u}(r,q) = 1. \qquad (5.5)$$

This is clear for $(\mathbf{1}, \mathbf{1})$ and for $(\mathbf{a}, \mathbf{a})$.

Concatenation. Let $(\mathbf{u}, \mathbf{u}_+)$ and $(\mathbf{v}, \mathbf{v}_+)$ be two extended limit-words satisfying (5.5). Consider two states r, q such that $(\mathbf{u}_+ \cdot \mathbf{v}_+)(r, q) = 1$ and r is $(\mathbf{u} \cdot \mathbf{v})$-recurrent.

Since r is $(\mathbf{u} \cdot \mathbf{v})$-recurrent, the recurrence class of r for $(\mathbf{u} \cdot \mathbf{v})$ is r itself, so $(\mathbf{u} \cdot \mathbf{v})(r, r) = 1$ and $(\mathbf{u} \cdot \mathbf{v})(r, p) = 1$ implies $p = r$. Let t be a state such that $\mathbf{u}(r, t) = 1$ and $\mathbf{v}(t, r) = 1$. Thanks to (5.3), rank$(t) =$ rank(r).

We argue that $\mathbf{u}(r, p) = 1$ implies $p = t$. Indeed, let p be a state such that $\mathbf{u}(r, p) = 1$. There exists a state p' such that $\mathbf{v}(p, p') = 1$, so in particular $(\mathbf{u} \cdot \mathbf{v})(r, p') = 1$, so $p' = r$. By (5.3), rank$(r) \leq$ rank$(p) \leq$ rank(r), so they are equal. By (5.4), since $\mathbf{u}(r, t) = 1$, $\mathbf{u}(r, p) = 1$ and rank$(r) =$ rank$(t) =$ rank(p), we have $p = t$. It follows that the state t is $\mathbf{u}$-recurrent: by Lemma 2.2, there exists a state p such that $\mathbf{u}(r, p) = 1$ and p is $\mathbf{u}$-recurrent. The above remark implies that $p = t$.

We argue that $\mathbf{v}(t, p) = 1$ implies $p = r$. Indeed, let p be a state such that $\mathbf{v}(t, p) = 1$. We have $(\mathbf{u} \cdot \mathbf{v})(r, p) = 1$, so $p = r$. It follows that the state r is $\mathbf{v}$-recurrent: by Lemma 2.2, there exists a state p such that $\mathbf{v}(t, p) = 1$ and p is $\mathbf{u}$-recurrent. The above remark implies that $p = r$.

Since $(\mathbf{u}_+ \cdot \mathbf{v}_+)(r, q) = 1$, there exists $s \in Q$ such that $\mathbf{u}_+(r, s) = 1$ and $\mathbf{v}_+(s, q) = 1$. By induction hypothesis for $(\mathbf{u}, \mathbf{u}_+)$, since $\mathbf{u}_+(r, s) = 1$ and r is $\mathbf{u}$-recurrent, we have $\mathbf{u}(r, s) = 1$.

Since r is $\mathbf{u}$-recurrent, its recurrence class is r itself, so $s = r$. By induction hypothesis for $(\mathbf{v}, \mathbf{v}_+)$, since $\mathbf{v}_+(r, q) = 1$ and r is $\mathbf{v}$-recurrent, we have $\mathbf{v}(r, q) = 1$.

It follows that $(\mathbf{u} \cdot \mathbf{v})(r, q) = 1$.

The case of iteration is easy.   □

The inclusion is strict, an example is given by figure 4.

5.4. **Leaktight automata strictly contain simple automata.** The class of simple automata has been defined in [CT12], where it was proved that the value 1 problem is decidable for a subset of this class (namely, for structurally simple automata). In the following we show that the class of simple automata is strictly contained in the class of leaktight automata.

We fix $\mathcal{A}$ a probabilistic automaton.

Definition 5.6 (Jets). *A jet is a sequence $(J_n)_{n \in \mathbb{N}}$ where $J_k \subseteq Q$ for each $k \in \mathbb{N}$.*

Definition 5.7 (Simple process [CT12]). *Let $w \in A^\omega$ be an infinite word. The process induced by w from the state p is simple if there exists $\lambda > 0$ and two jets $(A_n)_{n \in \mathbb{N}}$, $(B_n)_{n \in \mathbb{N}}$ such that:*

(1) for all $k \in \mathbb{N}$, A_k and B_k are disjoint and $A_k \cup B_k = Q$,

(2) for all $k \in \mathbb{N}$ and all $q \in A_k$, $\mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{<k}} q) \geq \lambda$,

(3) $\lim_{n \to \infty} \mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{<n}} B_n) = 0$.

Definition 5.8 (Simple automata [CT12]). *$\mathcal{A}$ is simple if for every infinite word w and every state p the process induced by w from p is simple.*

Theorem 5.9. *Every simple automaton is leaktight.*

The remainder of this subsection is devoted to the proof of Theorem 5.9. The proof is divided into two parts: first, we define non-simplicity witnesses, which are elements of the Markov monoid that witnesses the non-simplicity of an automaton, and second we show that if the Markov monoid of an automaton contains a leak, then it also contains a non-simplicity witness.

5.4.1. *Non-simplicity witness.*

Definition 5.10 (Non-simplicity witness). *A triple $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ of elements of the Markov monoid is a* non-simplicity *witness if there exist states r, t such that:*

(1) $\mathbf{u}\mathbf{v}^\sharp \mathbf{w}$ is idempotent,

(2) r is $\mathbf{u}\mathbf{v}^\sharp \mathbf{w}$-recurrent,

(3) $\mathbf{u}\mathbf{v}(r, t) = 1$,

(4) t is $\mathbf{v}$-transient.

Proposition 5.11. *If the Markov monoid of a probabilistic automaton contains a non-simplicity witness, then it is not simple.*

To prove Proposition 5.11, we rely on the following two lemmata:

Lemma 5.12. *Let $\mathbf{u}$ be an idempotent element of the Markov monoid, r be a state $\mathbf{u}$-recurrent, and $(u_n)_{n \in \mathbb{N}}$ a sequence of words that reifies $\mathbf{u}$. Then there exists a constant $\gamma > 0$ and a strictly increasing map $h : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, we have:*

$$\mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_{h(0)} \cdots u_{h(n-1)}} r) \geq \gamma.$$

Proof. Since the sequence of words $(u_n)_{n \in \mathbb{N}}$ reifies the limit-word $\mathbf{u}$, for all states s, t, $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t)$ converges and:

$$\mathbf{u}(s,t) = 1 \iff \lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) > 0. \tag{5.6}$$

Define:

$$\lambda = \frac{1}{2} \cdot \min\{\lim_n \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_n} t) \mid \mathbf{u}(s,t) = 1\}.$$

Thanks to (5.6), there exists an increasing map $h : \mathbb{N} \to \mathbb{N}$ such that the following two conditions hold, for all states s, t and for all $n \in \mathbb{N}$:

$$\text{if } \mathbf{u}(s,t) = 0, \text{ then } \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_{h(n)}} t) \leq \frac{1}{|Q| \cdot 2^{n+2}}, \tag{5.7}$$

$$\text{if } \mathbf{u}(s,t) = 1, \text{ then } \mathbb{P}_{\mathcal{A}}(s \xrightarrow{u_{h(n)}} t) \geq \lambda. \tag{5.8}$$

We now use (5.7) and (5.8) to prove the desired result, for $\gamma = \frac{\lambda}{2}$.

Let $w_n = u_{h(0)} u_{h(1)} \cdots u_{h(n-1)}$. Denote the $\mathbf{u}$-recurrence class of r by $R = \{q \in Q \mid \mathbf{u}(r,q) = 1\}$. We first bound the quantity $\mathbb{P}_{\mathcal{A}}(r \xrightarrow{w_n} Q \setminus R)$. Note that for $q \in R$ and t a state, the following holds: if $\mathbf{u}(q,t) = 1$ then $t \in R$, *i.e.* R is not left while following transitions consistent with $\mathbf{u}$. It follows that the probability to leave R from r while reading w_n is smaller than $\sum_{k=0}^{n-1} \mathbb{P}_{\mathcal{A}}(R \xrightarrow{u_{h(k)}} Q \setminus R)$, which is smaller than $\frac{1}{2}$ by (5.7). Thus $\mathbb{P}_{\mathcal{A}}(r \xrightarrow{w_{n-1}} R) \geq \frac{1}{2}$. Now, since r is $\mathbf{u}$-recurrent, and $\mathbf{u}$ is idempotent, for all $q \in R$ we have $\mathbf{u}(q,r) = 1$, so using (5.8) we get that $\mathbb{P}_{\mathcal{A}}(q \xrightarrow{u_{h(n-1)}} r) \geq \lambda$. It follows that $\mathbb{P}_{\mathcal{A}}(r \xrightarrow{w_n} r) \geq \frac{\lambda}{2}$, which concludes. $\square$

Lemma 5.13. *Let $w \in A^\omega$ be an infinite word. If there exist states p, s, t, $(i_n)_{n \in \mathbb{N}}$, $(j_n)_{n \in \mathbb{N}}$ and $\gamma > 0$ such that:*

(1) for all $n \in \mathbb{N}$, $\mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{<i_n}} s) \geq \gamma$,

(2) for all $n \in \mathbb{N}$, $i_n < j_n$ and $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{w[i_n, j_n]} t) > 0$,

(3) $\lim_{n \to \infty} \mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{<j_n}} t) = 0$,

then the process induced by w from p is not simple.

Proof. Assume towards contradiction that w induces a simple process from p with bound λ. We first argue that for infinitely many $n \in \mathbb{N}$, we have $s \in A_{i_n}$ and $t \in B_{j_n}$. Indeed, if this is not the case, then for $n \in \mathbb{N}$ large enough either $s \notin A_{i_n}$ or $t \notin B_{j_n}$, so either for infinitely many $n \in \mathbb{N}$ we have $s \notin A_{i_n}$, or for infinitely many $n \in \mathbb{N}$ we have $t \notin B_{j_n}$. The first case is contradicted by (1), the second case by (3).

Let $n \in \mathbb{N}$ such that $s \in A_{i_n}$ and $t \in B_{j_n}$, since $\mathbb{P}_{\mathcal{A}}(s \xrightarrow{w[i_n, j_n]} t) > 0$, along a path from s to t there is a transition from the jet A to the jet B. Formally, there exists k_n such that $i_n \leq k_n < j_n$ and the k_n^{th} transition goes from $q_{k_n} \in A_{k_n}$ to $q_{k_n+1} \in B_{k_n+1}$. This transition is a one-step transition in the automaton $\mathcal{A}$; denote by $p_{\min}$ the minimal non-zero probabilistic transition in $\mathcal{A}$, we have $\mathbb{P}_{\mathcal{A}}(q_{k_n} \xrightarrow{w[k_n, k_n+1]} q_{k_n+1}) \geq p_{\min}$. Now, consider

$\mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{\leq k_n}} q_{k_n+1})$; since $q_{k_n} \in A_{k_n}$, we have $\mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{<k_n}} q_{k_n}) \geq \lambda$, so $\mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{\leq k_n}} q_{k_n+1}) \geq \lambda \cdot p_{\min}$. But for infinitely many $n \in \mathbb{N}$, $q_{k_n+1} \in B_{k_n+1}$, contradicting $\lim_{n\to\infty} \mathbb{P}_{\mathcal{A}}(p \xrightarrow{w_{<n}} B_n) = 0$. □

Now we are ready to prove Proposition 5.11.

Proof. Let $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ be a non-simplicity witness, and $r, t \in Q$ such that r is $\mathbf{uv}^{\sharp}\mathbf{w}$-recurrent, $\mathbf{uv}(r,t) = 1$ and t is $\mathbf{v}$-transient. Let $(u_n)_{n\in\mathbb{N}}, (v_n)_{n\in\mathbb{N}}, (w_n)_{n\in\mathbb{N}}$ be sequences of words which reify $\mathbf{u}, \mathbf{v}, \mathbf{w}$ respectively.

Thanks to Proposition 2.11, there exists a strictly increasing map $f : \mathbb{N} \to \mathbb{N}$ such that $(v_{f(n)}^n)_{n\in\mathbb{N}}$ reifies $\mathbf{v}^{\sharp}$. Note that since $(v_{f(n)})_{n\in\mathbb{N}}$ is a subsequence of $(v_n)_{n\in\mathbb{N}}$, it also reifies $\mathbf{v}$. Since $\mathbf{v}$ is idempotent, for all $k \in \mathbb{N}$, $(v_{f(n)}^k)_{n\in\mathbb{N}}$ reifies $\mathbf{v}^k$. By assumption $\mathbf{uv}(r,t) = 1$, so there exists $N_k \in \mathbb{N}$ such that for all $n \geq N_k$, $\mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_n v_{f(n)}^k} t) > 0$.

Let $g(n) = \max(n, N_n)$. Since g is increasing, $(u_{g(n)} v_{f(g(n))}^{g(n)} w_{g(n)})_{n\in\mathbb{N}}$ is a subsequence of $(u_n v_{f(n)}^n w_n)_{n\in\mathbb{N}}$, so it reifies $\mathbf{uv}^{\sharp}\mathbf{w}$ as well. By definition of the function g, we have:

$$\text{for all } n \in \mathbb{N}, \quad \mathbb{P}_{\mathcal{A}}(r \xrightarrow{u_{g(n)} v_{f(g(n))}^n} t) > 0. \tag{5.9}$$

We apply Lemma 5.12 to the limit-word $\mathbf{uv}^{\sharp}\mathbf{w}$, the state r and the sequence of words $(u_{g(n)} v_{f(g(n))}^{g(n)} w_{g(n)})_{n\in\mathbb{N}}$, and obtain h a strictly increasing map and a constant $\gamma > 0$.

Define the new sequence of words $(z_n)_{n\in\mathbb{N}} = (u_{g(h(n))} \cdot v_{f(g(h(n)))}^{g(h(n))} \cdot w_{g(h(n))})_{n\in\mathbb{N}}$, and $(x_n)_{n\in\mathbb{N}} = (u_{g(h(n))} \cdot v_{f(g(h(n)))}^{h(n)})_{n\in\mathbb{N}}$. We have:

$$\text{for all } n \in \mathbb{N}, \quad \mathbb{P}_{\mathcal{A}}(r \xrightarrow{z_0 \cdots z_{n-1}} r) \geq \gamma. \tag{5.10}$$

Let $z = z_0 z_1 \cdots$. We argue that the conditions of Lemma 5.13 are met:
(1) for all $n \in \mathbb{N}$, $\mathbb{P}_{\mathcal{A}}(r \xrightarrow{z_0 \cdots z_{n-1}} r) \geq \gamma$,
(2) for all $n \in \mathbb{N}$, $\mathbb{P}_{\mathcal{A}}(r \xrightarrow{x_n} t) > 0$,
(3) $\lim_{n\to\infty} \mathbb{P}_{\mathcal{A}}(r \xrightarrow{z_0 \cdots z_{n-1} \cdot x_n} t) = 0$,

The item (1) is (5.10), the item (2) follows from (5.9), so we consider (3).

First, note that $(x_n)_{n\in\mathbb{N}}$ reifies $\mathbf{uv}^{\sharp}$. Indeed, we first argue that $(v_{f(g(n))}^n)_{n\in\mathbb{N}}$ reifies $\mathbf{v}^{\sharp}$: it follows from Proposition 2.11, since $f \circ g : \mathbb{N} \to \mathbb{N}$ is a strictly increasing map satisfying $f \circ g \geq f$, and that $(v_{f(n)}^n)_{n\in\mathbb{N}}$ reifies $\mathbf{v}^{\sharp}$. Now, $(v_{f(g(h(n)))}^{h(n)})_{n\in\mathbb{N}}$ is a subsequence of $(v_{f(g(n))}^n)_{n\in\mathbb{N}}$, so it reifies $\mathbf{v}^{\sharp}$ as well, and it follows that $(x_n)_{n\in\mathbb{N}}$ reifies $\mathbf{uv}^{\sharp}$.

Second, let $q \in Q$, since t is $\mathbf{v}$-transient, $\mathbf{uv}^{\sharp}(q,t) = 0$. Since $(x_n)_{n\in\mathbb{N}}$ reifies $\mathbf{uv}^{\sharp}$, we have $\lim_{n\to\infty} \mathbb{P}_{\mathcal{A}}(q \xrightarrow{x_n} t) = 0$. Now,

$$\mathbb{P}_{\mathcal{A}}(r \xrightarrow{z_0 \cdots z_{n-1} \cdot x_n} t) = \sum_{q \in Q} \mathbb{P}_{\mathcal{A}}(r \xrightarrow{z_0 \cdots z_{n-1}} q) \cdot \mathbb{P}_{\mathcal{A}}(q \xrightarrow{x_n} t),$$

and for each term in the sum, the second factor converges to zero, so $\lim_{n\to\infty} \mathbb{P}_{\mathcal{A}}(r \xrightarrow{z_0 \cdots z_{n-1} \cdot x_n} t) = 0$.

Thus Lemma 5.13 applies, and z induces a non-simple process from p, so $\mathcal{A}$ is not simple. □

5.4.2. *The presence of a leak implies the presence of a non-simplicity witness.* Now we show that the presence of a leak witness implies the presence of a non-simplicity witness.

Proposition 5.14. *If the extended Markov monoid of a probabilistic automaton contains a leak witness, then it also contains a non-simplicity witness.*

In the following proof, we will make use of the notion of $\sharp$-height for an extended limit-word. The $\sharp$-height of a limit-word was defined in Section 4.6, the $\sharp$-height of an extended limit-word $(\mathbf{u}, \mathbf{u}_+)$ is the $\sharp$-height of $\mathbf{u}$.

Proof. Let $\mathcal{A}$ be a probabilistic automaton whose extended Markov monoid contains a leak witness. Consider the subset $\mathcal{C}$ of extended limit-words $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid such that there exist states r, t satisfying:

(1) $(\mathbf{u}, \mathbf{u}_+)$ is idempotent,
(2) r is $\mathbf{u}$-recurrent,
(3) $\mathbf{u}(r, t) = 0$,
(4) $\mathbf{u}_+(r, t) = 1$.

Note that $\mathcal{C}$ is non-empty since every leak witness is in $\mathcal{C}$.

Consider an element $(\mathbf{z}, \mathbf{z}_+)$ in $\mathcal{C}$ of minimal $\sharp$-height and let $r, t \in Q$ such that r is $\mathbf{z}$-recurrent, $\mathbf{z}(r, t) = 0$ and $\mathbf{z}_+(r, t) = 1$. In particular, we have $\mathbf{z} \neq \mathbf{z}_+$, so in any decomposition of $(\mathbf{z}, \mathbf{z}_+)$ into concatenation and iteration there must be at least one iteration. Consequently, $\mathbf{z} = \mathbf{u}\mathbf{v}^{\sharp}\mathbf{w}$ for some $\mathbf{u}, \mathbf{v}, \mathbf{w}$, and $\mathbf{u}\mathbf{v}\mathbf{w}$ has a strictly smaller $\sharp$-height than $\mathbf{z}$.

Let R be the $\mathbf{z}$-recurrence class of r, and T the set of $\mathbf{v}$-transient states. We argue that the following holds:

$$\text{there exist } r' \in R \text{ and } t' \in T \text{ such that } \mathbf{u}\mathbf{v}(r', t') = 1. \tag{5.11}$$

Assume towards contradiction that (5.11) does not hold, *i.e.* for all $r' \in R$ and $t' \in T$, we have $\mathbf{u}\mathbf{v}(r', t') = 0$, then we prove that $(\mathbf{u}\mathbf{v}\mathbf{w}, \mathbf{z}_+)^{|\mathcal{G}_+|!}$ is in $\mathcal{C}$, contradicting the minimality of $(\mathbf{z}, \mathbf{z}_+)$ as it has strictly smaller $\sharp$-height.

First observe that for all states q we have $\mathbf{u}\mathbf{v}(r', q) = \mathbf{u}\mathbf{v}^{\sharp}(r', q)$, which implies that for all $r' \in R$ and state q, we have:

$$\mathbf{u}\mathbf{v}\mathbf{w}(r', q) = \mathbf{u}\mathbf{v}^{\sharp}\mathbf{w}(r', q) = \mathbf{z}(r', q). \tag{5.12}$$

We check that $(\mathbf{u}\mathbf{v}\mathbf{w}, \mathbf{z}_+)^{|\mathcal{G}_+|!}$ is in $\mathcal{C}$, with the states r, t as witnesses:

(1) $(\mathbf{u}\mathbf{v}\mathbf{w}, \mathbf{z}_+)^{|\mathcal{G}_+|!}$ is idempotent, this follows from Lemma 2.2.
(2) r is $(\mathbf{u}\mathbf{v}\mathbf{w})^{|\mathcal{G}_+|!}$-recurrent. Indeed, let $q \in Q$ such that $(\mathbf{u}\mathbf{v}\mathbf{w})^{|\mathcal{G}_+|!}(r, q) = 1$. It follows that $(\mathbf{u}\mathbf{v}^{\sharp}\mathbf{w})^{|\mathcal{G}_+|!}(r, q) = 1$, but $(\mathbf{u}\mathbf{v}^{\sharp}\mathbf{w})^{|\mathcal{G}_+|!} = \mathbf{z}^{|\mathcal{G}_+|!} = \mathbf{z}$, so $\mathbf{z}(r, q) = 1$. Since r is $\mathbf{z}$-recurrent, we have $\mathbf{z}(q, r) = 1$ and $q \in R$, so (5.12) implies that $\mathbf{z}(q, r) = \mathbf{u}\mathbf{v}\mathbf{w}(q, r)$, thus $\mathbf{u}\mathbf{v}\mathbf{w}(q, r) = 1$. Also, $\mathbf{z}(r, r) = \mathbf{u}\mathbf{v}\mathbf{w}(r, r)$, so $\mathbf{u}\mathbf{v}\mathbf{w}(r, r) = 1$, and altogether $(\mathbf{u}\mathbf{v}\mathbf{w})^{|\mathcal{G}_+|!}(q, r) = 1$.
(3) $(\mathbf{u}\mathbf{v}\mathbf{w})^{|\mathcal{G}_+|!}(r, t) = 0$. Indeed, assume towards contradiction that $(\mathbf{u}\mathbf{v}\mathbf{w})^{|\mathcal{G}_+|!}(r, t) = 1$. Then there exist $q_0, q_1, \ldots, q_{|\mathcal{G}_+|!}$ such that $q_0 = r$, for $i \in \{0, \ldots, |\mathcal{G}_+|! - 1\}$ we have $\mathbf{u}\mathbf{v}\mathbf{w}(q_i, q_{i+1}) = 1$ and $q_{|\mathcal{G}_+|!} = t$. We prove by induction on $i \in \{0, \ldots, |\mathcal{G}_+|! - 1\}$ that q_i is in R and that $\mathbf{z}(q_i, q_{i+1}) = 1$. Assume q_i is in R, then by (5.12), $\mathbf{u}\mathbf{v}\mathbf{w}(q_i, q_{i+1}) = \mathbf{z}(q_i, q_{i+1})$, so $\mathbf{z}(q_i, q_{i+1}) = 1$. But q_i is in R, which is a recurrence class for $\mathbf{z}$, so q_{i+1} is in R as well, concluding the induction. Thus, we have $\mathbf{z}^{|\mathcal{G}_+|!}(r, t) = 1$, and since $\mathbf{z}$ is idempotent $\mathbf{z}(r, t) = 1$, a contradiction.
(4) $\mathbf{z}_+^{|\mathcal{G}_+|!}(r, t) = 1$. It follows from the fact the $\mathbf{z}_+$ is idempotent, and that $\mathbf{z}_+(r, t) = 1$.

We reached a contradiction, since $(\mathbf{uvw}, \mathbf{z}_+)^{|\mathcal{G}_+|!}$ has a strictly smaller $\sharp$-height than $(\mathbf{z}, \mathbf{z}_+)$. It follows from (5.11) that $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ is a non-simplicity witness for the states r' and t', concluding the proof. $\qquad\square$

The proof of Theorem 5.9 is now a simple combination of Proposition 5.11 and Proposition 5.14. The inclusion is strict: figure 4 provides an example of a leaktight automaton which is not simple.

6. Probabilistic ω-automata

In this section, we relate the value 1 problem for probabilistic automata over finite words and over infinite words, as introduced and studied in [BBG12]. We state a general theorem, showing the equivalence of the value 1 problem for automata over finite words and the value 1 problem for automata over infinite words with the parity condition. This theorem allows to extend the decidability results from finite words to infinite ones.

For the definitions of probabilistic automata over infinite words, we refer to [BBG12]. For the sake of readability, we introduce two notations in this section. First, we denote by $\mathbb{P}_{\mathcal{A},w}(E)$ the probability of the measurable event E when reading the infinite word w on $\mathcal{A}$. Second, we denote by $\mathbb{P}_{\mathcal{A}}^\delta(u)$ the probability that the finite word u is accepted by $\mathcal{A}$ with δ as initial distribution. Note that in our definition, probabilistic automata have a unique initial state; however here we need to deal with automata having general initial distributions, so we sometimes consider the more general tuples $\mathcal{A} = (Q, \delta, \Delta, F)$ where δ is the initial probability distribution.

Theorem 6.1. *Let $\mathcal{A} = (Q, \delta_0, \Delta, c)$ be a probabilistic parity automaton where $c : Q \to \mathbb{N}$ is a priority function. Consider the deterministic transducer $\mathcal{M}$ over $\mathcal{A}$, which keeps track of the minimal priority seen: $\mathcal{M} = (c(Q), Q^\mathcal{A}, \Delta_\mathcal{M})$ where $\Delta_\mathcal{M}(q, c) = \min(c, c(q))$.*

The automaton $\mathcal{A}$ over infinite words has value 1 if and only if there exists $R \subseteq Q$, such that the two following probabilistic automata over finite words have value 1:

- *The automaton $\mathcal{A}(R)$ with R as set of final states;*
- *The automaton $\mathcal{A} \otimes \mathcal{M}$ with the uniform distribution over $R_c = \{(q, c(q)) \mid q \in R\}$ as initial distribution and $\{(q, e) \mid q \in R \text{ and } e \text{ even}\}$ as set of final states.*

Before giving the proof, we need two lemma.

Lemma 6.2. *Let $\mathcal{A} = (Q, \delta, \Delta, F)$ be a probabilistic automaton, u a word and $\varepsilon > 0$. Let $\mu = \min\{\delta(q) \mid q \in \operatorname{Supp}(\delta)\}$. If $\mathbb{P}_{\mathcal{A}}^\delta(u) \geq 1 - \varepsilon \cdot \mu$, then for all $q \in \operatorname{Supp}(\delta)$, $\mathbb{P}_{\mathcal{A}}^q(u) \geq 1 - \varepsilon$.*

Proof.
$$\mathbb{P}_{\mathcal{A}}^\delta(u) = \sum_{q \in \operatorname{Supp}(\delta)} \delta(q) \cdot \mathbb{P}_{\mathcal{A}}^q(u) \ \geq \ 1 - \varepsilon \cdot \mu.$$
Let $q \in \operatorname{Supp}(\delta)$, since the probabilities are bounded by 1:
$$\delta(q) \cdot \mathbb{P}_{\mathcal{A}}^q(u) \ + \ \underbrace{\sum_{p \in \operatorname{Supp}(\delta),\ p \neq q} \delta(p)}_{1 - \delta(q)} \ \geq \ 1 - \varepsilon \cdot \mu.$$
Hence:
$$\mathbb{P}_{\mathcal{A}}^q(u) \ \geq \ 1 - \varepsilon \cdot \frac{\mu}{\delta(q)} \ \geq \ 1 - \varepsilon.$$

$\square$

Corollary 6.3. *Let $\mathcal{A} = (Q, \delta, \Delta, F)$ be a probabilistic automaton. If $\mathcal{A}$ has value 1, then:*

- *For all distributions δ' such that $\mathrm{Supp}(\delta') \subseteq \mathrm{Supp}(\delta)$, it has value 1 with δ' as initial distribution.*
- *For all distributions δ' such that $\sum_{q \in \mathrm{Supp}(\delta)} \delta'(q) \geq \theta$, it has value at least θ with δ' as initial distribution.*

Proof. Assume $\mathcal{A}$ has value 1. Let $\varepsilon > 0$, then there exists a word u such that $\mathbb{P}_{\mathcal{A}}(u) \geq 1 - \varepsilon \cdot \mu$. Thanks to Lemma 6.2, this implies that for all $q \in \mathrm{Supp}(\delta)$, we have $\mathbb{P}_{\mathcal{A}}^q(u) \geq 1 - \varepsilon$.
Now for δ':

$$\mathbb{P}_{\mathcal{A}}^{\delta'}(u) \;=\; \sum_{q \in Q} \delta'(q) \cdot \mathbb{P}_{\mathcal{A}}^q(u) \;\geq\; \sum_{q \in \mathrm{Supp}(\delta)} \delta'(q) \cdot \mathbb{P}_{\mathcal{A}}^q(u) \;\geq\; \left(\sum_{q \in \mathrm{Supp}(\delta)} \delta'(q) \right) \cdot (1 - \varepsilon).$$

For the first item, note that if $\mathrm{Supp}(\delta') \subseteq \mathrm{Supp}(\delta)$, then $\sum_{q \in \mathrm{Supp}(\delta)} \delta'(q) = 1$, so $\mathbb{P}_{\mathcal{A}}^{\delta'}(u) \geq 1 - \varepsilon$, so $\mathcal{A}$ has value 1 with δ' as initial distribution.

The second item follows from the last inequality, implying $\mathbb{P}_{\mathcal{A}}^{\delta'}(u) \geq \theta \cdot (1 - \varepsilon)$, so $\mathcal{A}$ has value at least θ with δ' as initial distribution. $\square$

Lemma 6.4. *For all $\varepsilon > 0$, there exists a sequence $(\varepsilon_k)_{k \geq 0}$ satisfying:*

(1) *For all $k \geq 0$, we have $0 < \varepsilon_k < 1$;*
(2) *$\prod_{k \geq 0} \varepsilon_k \geq 1 - \varepsilon$;*
(3) *For all $k \geq 0$, we have $\prod_{p \leq k} \varepsilon_p > \varepsilon_{k+1}$.*

Proof. Define $\nu_k = 1 - \frac{1}{2^{2^k}}$, which clearly satisfies 1. It also satisfies 3.:

$$\begin{aligned} \prod_{p \leq k} \nu_p &= \prod_{p \leq k} \frac{2^{2^p} - 1}{2^{2^p}} \\ &= \frac{\prod_{p \leq k} 2^{2^p} - 1}{2^{2^{k+1} - 1}} \end{aligned}$$

The inequality $\prod_{p \leq k} \nu_p > \nu_{k+1}$ is equivalent to:

$$2 \cdot \left(\prod_{p \leq k} (2^{2^p} - 1) \right) < 2^{2^{k+1}} - 1,$$

which is easily proved by induction, since $(2^{2^{k+1}} - 1)^2 < 2^{2^{k+2}} - 1$.

The infinite product $\prod_{k \geq 0} \nu_k$ has a value $\nu \approx 0.350184$, in particular $0 < \nu < 1$. Let $\varepsilon > 0$, then for $\lambda = \frac{\ln(1-\varepsilon)}{\ln(\nu)}$, which satisfies $\lambda > 0$, the sequence $(\nu_k^\lambda)_{k \in \mathbb{N}}$ satisfies the three conditions. $\square$

We are now fully equipped for the proof of Theorem 6.1

We begin with the left-to-right direction. Let $(w_n)_{n \in \mathbb{N}}$ be a sequence of infinite words such that $\mathbb{P}_{\mathcal{A}, w_n}(\mathrm{Parity}(c))$ converges to 1. For each n, denote by $(\delta_k^n)_{k \in \mathbb{N}}$ the sequence of distributions assumed by $\mathcal{A}$ when reading w_n. Since there are finitely many possible supports, there exists a subsequence where all distributions have the same support. Since $[0,1]^Q$ is a compact space, we can extract from this subsequence another one, denoted by $(\delta_{\phi_n(k)}^n)_{k \in \mathbb{N}}$, which converges to a distribution denoted by δ^n. By the same compactness

argument, at the expense of considering a subsequence of $(w_n)_{n \in \mathbb{N}}$, we can assume that $(\delta^n)_{n \in \mathbb{N}}$ converges to a distribution δ, whose support is denoted by R.

We now prove that $\mathcal{A}(R)$ as well as $\mathcal{A} \otimes \mathcal{M}$ have value 1.

The automaton $\mathcal{A}(R)$. Let $\varepsilon > 0$. Since $(\delta^n)_{n \in \mathbb{N}}$ converges to δ, for n large enough, we have $||\delta^n - \delta||_\infty \leq \frac{\varepsilon}{2 \cdot |R|}$. We fix n large enough; since $(\delta^n_{\phi_n(k)})_{k \in \mathbb{N}}$ converges to δ^n, for k large enough, we have $||\delta^n_{\phi_n(k)} - \delta^n||_\infty \leq \frac{\varepsilon}{2 \cdot |R|}$. We fix such a large k, altogether this implies $||\delta^n_{\phi_n(k)} - \delta||_\infty \leq \frac{\varepsilon}{|R|}$.

Now consider u the prefix of w_n until position $\phi_n(k)$. We have:

$$\mathbb{P}_{\mathcal{A}(R)}(u) \ = \ \sum_{q \in R} \delta^n_{\phi_n(k)}(q) \ \geq \ \sum_{q \in R} \left(\delta(q) - \frac{\varepsilon}{|R|} \right) \ = \ 1 - \varepsilon.$$

It follows that $\mathcal{A}(R)$ has value 1.

The automaton $\mathcal{A} \otimes \mathcal{M}$. Let $\nu = \min\{\delta(q) \mid q \in \mathrm{Supp}(\delta)\}$, clearly $0 < \nu < 1$. Since $(\delta^n)_{n \in \mathbb{N}}$ converges to δ, we can assume that for all $n \geq 0$, $||\delta^n - \delta||_\infty \leq \frac{\nu}{4}$, considering the sequence $(w_n)_{n \in \mathbb{N}}$ from some index on. For each n, the sequence $(\delta^n_{\phi_n(k)})_{k \in \mathbb{N}}$ converges to δ^n, so again by considering the sequence from some index on, we can assume that for all $k \geq 0$, $||\delta^n_{\phi_n(k)} - \delta^n||_\infty \leq \frac{\nu}{4}$, hence $||\delta^n_{\phi_n(k)} - \delta||_\infty \leq \frac{\nu}{2}$. As a result, for all $q \in R$, we have $\delta^n_{\phi_n(k)}(q) \geq \frac{\nu}{2} > 0$, so $R = \mathrm{Supp}(\delta) \subseteq \mathrm{Supp}(\delta^n) \subseteq \mathrm{Supp}(\delta^n_{\phi_n(k)})$.

Let $\varepsilon > 0$, and n such that $\mathbb{P}_{\mathcal{A},w_n}(\mathrm{Parity}(c)) \geq 1 - \varepsilon \cdot \frac{\nu}{4}$.

Let $\mathrm{Büchi}(d)$ be the set of runs where the color d is reached infinitely often and by $\mathrm{CoBüchi}(< d)$ the set of runs where colors less than d are reached only finitely often. Observe that:

$$\mathrm{Parity}(c) \ = \ \biguplus_{d \text{ even color}} \mathrm{Büchi}(d) \cap \mathrm{CoBüchi}(< d).$$

Let $\mathrm{Safe}(< d, k)$ be the set of runs where colors less than d are not reached anymore after the position $\phi_n(k)$. Observe that:

$$\mathrm{Büchi}(d) \cap \mathrm{CoBüchi}(< d) \ \subseteq \ \mathrm{Büchi}(d) \cap \bigcup_{k \geq 0} \mathrm{Safe}(< d, k),$$

so:

$$\mathbb{P}_{\mathcal{A},w_n}\left(\mathrm{Büchi}(d) \cap \mathrm{CoBüchi}(< d)\right) \ \leq \ \mathbb{P}_{\mathcal{A},w_n}\left(\mathrm{Büchi}(d) \cap \bigcup_{k \geq 0} \mathrm{Safe}(< d, k) \right).$$

Since the sequence $\mathrm{Büchi}(d) \cap (\mathrm{Safe}(< d, k))_{k \geq 0}$ is increasing with respect to set inclusion, for some k we have:

$$\mathbb{P}_{\mathcal{A},w_n}(\mathrm{Büchi}(d) \cap \mathrm{Safe}(< d, k)) \ \geq \ \mathbb{P}_{\mathcal{A},w_n}(\mathrm{Büchi}(d) \cap \mathrm{CoBüchi}(< d)) - \varepsilon \cdot \frac{\nu}{4}.$$

Note that here, k depends on d; however, since if it holds for k then it holds for any bigger k and that there are finitely many even colors d, we can assume that k is uniform over all even colors d.

Now, let $\mathrm{Reach}(d, k, k')$ be the set of runs where the color d is reached between the positions $\phi_n(k)$ and $\phi_n(k')$. Observe that:

$$\mathrm{Büchi}(d) \cap \mathrm{Safe}(< d, k) \ \subseteq \ \bigcup_{k' > k} \mathrm{Reach}(d, k, k') \cap \mathrm{Safe}(< d, k),$$

so:

$$\mathbb{P}_{\mathcal{A},w_n}(\text{Büchi}(d) \cap \text{Safe}(< d, k)) \ \leq \ \mathbb{P}_{\mathcal{A},w_n}\left(\bigcup_{k'>k} \text{Reach}(d,k,k') \cap \text{Safe}(< d, k)\right).$$

Since the sequence $(\text{Reach}(d,k,k') \cap \text{Safe}(< d, k))_{k'>k}$ is increasing with respect to set inclusion, for some k' we have:

$$\mathbb{P}_{\mathcal{A},w_n}(\text{Reach}(d,k,k') \cap \text{Safe}(< d, k)) \ \geq \ \mathbb{P}_{\mathcal{A},w_n}(\text{Büchi}(d) \cap \text{Safe}(< d, k)) - \varepsilon \cdot \frac{\nu}{4}.$$

Here again, k' depends on d; however with the same reasoning as above, we can assume that k' is uniform over all even colors d.

Altogether, this implies:

$$\mathbb{P}_{\mathcal{A},w_n}(\text{Reach}(d,k,k') \cap \text{Safe}(< d, k)) \ \geq \ \mathbb{P}_{\mathcal{A},w_n}(\text{Büchi}(d) \cap \text{CoBüchi}(< d)) - \varepsilon \cdot \frac{\nu}{2},$$

and summing these equalities for each even color d:

$$\begin{aligned}
\mathbb{P}_{\mathcal{A},w_n}(\text{Parity}(c,k,k')) \ &\geq \ \textstyle\sum_{d \text{ even color}} \mathbb{P}_{\mathcal{A},w_n}(\text{Reach}(d,k,k') \cap \text{Safe}(< d, k)) \\
&\geq \ \mathbb{P}_{\mathcal{A},w_n}(\text{Parity}(c)) - \varepsilon \cdot \tfrac{\nu}{2},
\end{aligned}$$

where $\text{Parity}(c,k,k')$ is the set of runs where the minimal color seen between the positions $\phi_n(k)$ and $\phi_n(k')$ is even.

Let v be the infix of w_n between the positions $\phi_n(k)$ and $\phi_n(k')$. The distribution $\delta^n_{\phi_n(k)}$ is over the set of states Q; we embed it as $\widehat{\delta}$ over the set of states $Q \times c(Q)$, setting $\widehat{\delta}(q, c(q)) = \delta^n_{\phi_n(k)}(q)$ and 0 otherwise. By construction of v, we have $\mathbb{P}^{\widehat{\delta}}_{\mathcal{A} \otimes \mathcal{M}}(v) = \mathbb{P}_{\mathcal{A},w_n}(\text{Parity}(c,k,k')) \geq 1 - \varepsilon \cdot \frac{\nu}{2}$. However, the initial distribution of $\mathcal{A} \otimes \mathcal{M}$ is uniform over R_c. We apply Lemma 6.2; by construction $\mu \geq \frac{\nu}{2}$, so $\mathbb{P}^{\widehat{\delta}}_{\mathcal{A} \otimes \mathcal{M}}(v) \geq 1 - \varepsilon \cdot \mu$. It follows that for all $q \in \text{Supp}(\delta^n_{\phi_n(0)})$, $\mathbb{P}^{q,c(q)}_{\mathcal{A} \otimes \mathcal{M}}(v) \geq 1 - \varepsilon$. Since $R \subseteq \text{Supp}(\delta^n_{\phi_n(k)})$, we have $\mathbb{P}_{\mathcal{A} \otimes \mathcal{M}}(v) \geq 1 - \varepsilon$, hence $\mathcal{A} \otimes \mathcal{M}$ has value 1.

We now prove the right-to-left direction. Let R such that both $\mathcal{A}(R)$ and $\mathcal{A} \otimes \mathcal{M}$ have value 1. Let $\varepsilon > 0$, consider the sequence $(\varepsilon_k)_{k \geq 0}$ given by Lemma 6.4.

Since $\mathcal{A}(R)$ has value 1, there exists u such that $\mathbb{P}_{\mathcal{A}(R)}(u) \geq \varepsilon_0$. We show the existence of a sequence $w_1, w_2, \ldots$ of words such that for all $k \geq 0$:

$$\mathbb{P}_{\mathcal{A}(R)}(u \cdot w_1 \cdot w_2 \cdot \ldots \cdot w_k) \geq \prod_{p \leq k} \varepsilon_p \qquad \text{and} \qquad \mathbb{P}^{\delta'_k}_{\mathcal{A} \otimes \mathcal{M}}(w_{k+1}) \geq \varepsilon_{k+1},$$

where δ_k is the distribution obtained by reading $u \cdot w_1 \cdot w_2 \cdot \ldots \cdot w_k$ on $\mathcal{A}$; the distribution δ_k is over the set of states Q, we embed it as δ'_k over the set of states $Q \times c(Q)$, setting $\delta'_k(q, c(q)) = \delta_k(q)$ and 0 otherwise.

We proceed inductively; assume $w_1, \ldots, w_k$ have been chosen. Note that by construction, $\sum_{q \in R} \delta_k(q) \geq \prod_{p \leq k} \varepsilon_p$. However, the support of δ_k is not included in R, so the first item of Corollary 6.3 does not apply. Then $\sum_{q \in R_c} \delta'_k(q) \geq \prod_{p \leq k} \varepsilon_p$.

Since $\mathcal{A} \otimes \mathcal{M}$ has value 1, thanks to the second item of Corollary 6.3, it has value at least $\prod_{p \leq k} \varepsilon_p$ for δ'_k as initial distribution. Together with 3., this implies that there exists w_{k+1} such that $\mathbb{P}^{\delta'_k}_{\mathcal{A} \otimes \mathcal{M}}(w_{k+1}) \geq \varepsilon_{k+1}$.

We have:

$$
\begin{aligned}
\mathbb{P}_{\mathcal{A}(R)}(u \cdot w_1 \cdot w_2 \cdot \ldots \cdot w_{k+1}) \ &\geq\ \mathbb{P}_{\mathcal{A}(R)}(u \cdot w_1 \cdot w_2 \cdot \ldots \cdot w_k) \cdot \mathbb{P}_{\mathcal{A}(R)}^{\delta_k}(w_{k+1}) \\
&\geq\ \mathbb{P}_{\mathcal{A}(R)}(u \cdot w_1 \cdot w_2 \cdot \ldots \cdot w_k) \cdot \mathbb{P}_{\mathcal{A} \otimes \mathcal{M}}^{\delta_k'}(w_{k+1}) \\
&\geq\ \left(\textstyle\prod_{p \leq k} \varepsilon_p \right) \ \cdot\ \varepsilon_{k+1} \\
&=\ \textstyle\prod_{p \leq k+1} \varepsilon_p,
\end{aligned}
$$

which concludes the inductive construction.

Let $w = u \cdot w_1 \cdot w_2 \cdot \ldots$. We evaluate $\mathbb{P}_{\mathcal{A},w}(\mathrm{Parity}(c))$:

$$
\mathbb{P}_{\mathcal{A},w}(\mathrm{Parity}(c)) \ \geq\ \mathbb{P}_{\mathcal{A}(R)}(u) \cdot \prod_{k \geq 0} \mathbb{P}_{\mathcal{A} \otimes \mathcal{M}}^{\delta_k'}(w_{k+1}) \ \geq\ \varepsilon_0 \cdot \prod_{k \geq 1} \varepsilon_k \ =\ \prod_{k \geq 0} \varepsilon_k \ \geq\ 1 - \varepsilon.
$$

It follows that $\mathcal{A}$ has value 1 as a probabilistic parity automaton, which concludes the proof of Theorem 6.1.

Corollary 6.5. *The value 1 problem for leaktight automata over infinite words with the parity condition is decidable and* PSPACE-*complete.*

Indeed, observe that if $\mathcal{A}$ is leaktight, then by Proposition 5.4, so is $\mathcal{A} \otimes \mathcal{M}$.

Acknowledgment

Conclusion

We introduced a subclass of probabilistic automata, called leaktight automata, for which we proved that the value 1 problem is PSPACE-complete. This subclass generalizes all subclasses of probabilistic automata whose value 1 problem is known to be decidable.

A challenging perspective is now to find subclasses of partially observable Markov decision processes where the value 1 problem is decidable (some preliminary results were given in [GO14]), and to extend our results to the setting of partially observable stochastic games, which is even more challenging.

References

[BBG12]   Christel Baier, Nathalie Bertrand, and Marcus Größer. Probabilistic ω-automata. *Journal of the ACM*, 59(1):1, 2012.

[Ber74]   Alberto Bertoni. The solution of problems relative to probabilistic automata in the frame of the formal languages theory. In *GI Jahrestagung*, pages 107–112, 1974.

[CDHR07] Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, 3(3), 2007.

[CK97]   Karel Culik and Jarkko Kari. *Digital images and formal languages*. Springer-Verlag New York, Inc., 1997.

[CMRR08] Corinna Cortes, Mehryar Mohri, Ashish Rastogi, and Michael Riley. On the computation of the relative entropy of probabilistic automata. *International Journal of Foundations of Computer Science*, 19(1):219–242, 2008.

[Col09]   Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *ICALP (2)*, pages 139–150, 2009.

[Col11]   Thomas Colcombet. Green's relations and their use in automata theory. In *LATA*, pages 1–21, 2011.

[CSV11]   Rohit Chadha, A. Prasad Sistla, and Mahesh Viswanathan. Power of randomization in automata on infinite strings. *Logical Methods in Computer Science*, 7(3), 2011.

[CSV13]   Rohit Chadha, A. Prasad Sistla, and Mahesh Viswanathan. Probabilistic automata with isolated cut-points. In *MFCS*, pages 254–265, 2013.

[CT12]   Krishnendu Chatterjee and Mathieu Tracol. Decidable problems for probabilistic automata on infinite words. In *LICS*, pages 185–194, 2012.

[DEKM99]   Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

[FGO12]   Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. In *LICS*, pages 295–304, 2012.

[Fij14]   Nathanaël Fijalkow. What is known about the value 1 problem for probabilistic automata? *CoRR*, abs/1410.3770, 2014.

[GO10]   Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *ICALP (2)*, pages 527–538, 2010.

[GO14]   Hugo Gimbert and Youssouf Oualhadj. Deciding the value 1 problem for sharp-acyclic partially observable markov decision processes. In *SOFSEM*, pages 281–292, 2014.

[How95]   John M. Howie. *Fundamentals of semigroup theory*. Clarendon Press, Oxford, 1995.

[Kir05]   Daniel Kirsten. Distance desert automata and the star height problem. *ITA*, 39(3):455–509, 2005.

[Koz77]   Dexter Kozen. Lower bounds for natural proof systems. In *FOCS*, pages 254–266, 1977.

[Lal79]   Gérard Lallement. *Semigroups and Combinatorial Applications*. Wiley, 1979.

[Moh97]   Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:269–311, June 1997.

[Paz71]   Azaria Paz. *Introduction to probabilistic automata*. Academic Press, 1971.

[Rab63]   Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[Sch61]   Marcel Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.

[Sim90]   Imre Simon. Factorization forests of finite height. *Theoretical Computer Science*, 72(1):65–94, 1990.

[Sim94]   Imre Simon. On semigroups of matrices over the tropical semiring. *ITA*, 28(3-4):277–294, 1994.

[Tor11]   Szymon Toruńczyk. *Languages of profinite words and the limitedness problem*. PhD thesis, University of Warsaw, 2011.

[Tze92]   Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM Journal of Computation*, 21(2):216–227, 1992.