Deciding the Value 1 Problem of Probabilistic Leaktight Automata

Nathanaël Fijalkow, Hugo Gimbert, Youssouf Oualhadj

Deciding the Value 1 Problem of Probabilistic Leaktight Automata

Nathanaël Fijalkow[*] Hugo Gimbert[†] and Youssouf Oualhadj[‡]

October 25, 2011

Abstract

The value 1 problem is a decision problem for probabilistic automata over finite words: given a probabilistic automaton $\mathcal{A}$, are there words accepted by $\mathcal{A}$ with probability arbitrarily close to 1? This problem was proved undecidable recently, even in restricted cases [FGO11, GO10]. We introduce a new class of probabilistic automata, called *leaktight automata*, for which the value 1 problem is shown decidable (and PSPACE-complete). We construct an algorithm based on the computation of a monoid abstracting the behaviours of the automaton, and rely on algebraic techniques developed by Simon for the correctness proof. The class of leaktight automata is decidable in PSPACE, subsumes all subclasses of probabilistic automata whose value 1 problem is known to be decidable (in particular deterministic automata), and is closed under two natural composition operators.

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 Q and reads input words over a finite alphabet A. The computation starts from the initial state i 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 to terminate the computation in one of the final states $F \subseteq Q$.

From a language-theoretic perspective, several algorithmic properties of probabilistic automata are known: while language emptiness is undecidable [Ber74, GO10, Paz71], language equivalence is decidable [CMR07, Sch61, Tze92] as well as other properties [CL89, CMRR08].

Rather than formal language theory, 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.

[*]LIAFA, Université Denis Diderot - Paris 7, France. nath@liafa.jussieu.fr
[†]LaBRI, CNRS, Bordeaux, France. hugo.gimbert@labri.fr
[‡]LaBRI, Université Bordeaux 1, France. youssouf.oualhadj@labri.fr

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 [1] :

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

Recently, the second and third authors of the present paper proved that the value 1 problem is undecidable [GO10]. However, probabilistic automata, and more generally partially observable Markov decision processes and stochastic games, are a widely used model of probabilistic machines used in many fields like software verification [BBG08, 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.

Our result. Following this goal, we found a new class of probabilistic automata, *leaktight automata*, for which the value 1 problem is decidable. This subclass subsumes all known subclasses of probabilistic automata sharing this decidability property and is closed under parallel composition and synchronized product. 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. The value 1 problem was proved decidable for a sub-class of probabilistic automata called $\sharp$-acyclic automata [GO10]. Since the class of $\sharp$-acyclic automata is strictly contained in the class of leaktight automata, the result of the present paper extends the decidability result of [GO10]. Chadha et al. [CSV09] recently introduced the class of hierarchical probabilistic automata, which is also strictly contained in the class of leaktight automata. As a consequence of our result, the value 1 problem is decidable for hierarchical probabilistic automata. Our proof techniques totally depart from the ones used in [CSV09, GO10]. Instead, 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].

Outline. Basic definitions are given in Section 1, our algorithm is presented in Section 2, the decidability of the leaktight property is given in Section 3 and finally Section 4 investigates properties and provides examples of leaktight automata. The proofs can be found in the appendix.

1 Definitions

1.1 Probabilistic automata

Let Q be a finite set of states. A probability distribution over Q is a row vector δ of size $|Q|$ whose coefficients are in $[0,1]$ such that $\sum_{q\in Q}\delta(q)=1$. A probabilistic transition matrix M is a square matrix in $[0,1]^{Q\times Q}$ such that every row of M is a probability distribution over Q.

Definition 1 (Probabilistic automata)**.** *A tuple $\mathcal{A}=(Q,A,(M_a)_{a\in A},i,F)$ represents a probabilistic automaton, where Q is a finite set of states, A is the finite input alphabet, $(M_a)_{a\in A}$ are*

[1] Bertoni formulated the value 1 problem in a different yet equivalent way: "Is the cut-point 1 isolated or not?".

the probabilistic transition matrices, $i \in Q$ is the initial state and $F \subseteq Q$ is the set of accepting states.

For each letter $a \in A$, $M_a(s,t)$ is the probability to go from state s to state t when reading letter a. Given an input word $w \in A^*$, we denote $0 \leq w(s,t) \leq 1$ the probability to go from state s to state t when reading the word w. Formally, if $w = a_1 a_2 \cdots a_n$ then $w(s,t) = (M_{a_1} M_{a_2} \cdots M_{a_n})(s,t)$.

Definition 2 (Value and acceptance probability)**.** *The* acceptance probability *of a word $w \in A^*$ by $\mathcal{A}$ is $\mathbb{P}_{\mathcal{A}}(w) = \sum_{f \in F} w(i,f)$. The* value *of $\mathcal{A}$, denoted $val(\mathcal{A})$, is the supremum acceptance probability over all possible input words:*

$$val(\mathcal{A}) = \sup_{w \in A^*} \mathbb{P}_{\mathcal{A}}(w) \ . \tag{1}$$

1.2 The value 1 problem for probabilistic automata

We are interested in the following decision problem:

Problem 1 (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.

1.2.1 Parallel convergence rates and undecidability

Whereas the formulation of the value 1 problem only relies *qualitatively* on the asymptotic behaviour of probabilities (the probability to be in non-final states should be arbitrarily small) the answer to the value 1 problem depends *quantitatively* on the transition probabilities.

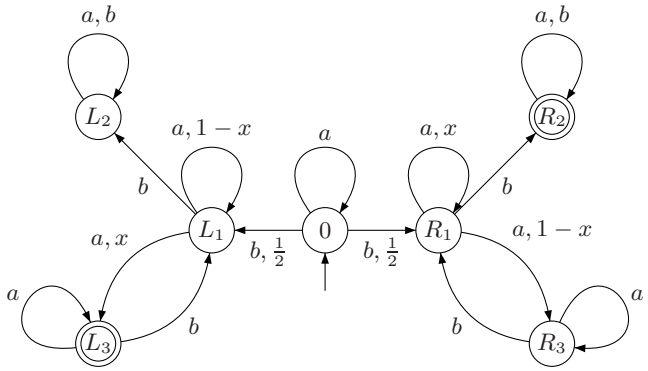

Figure 1: This automaton has value 1 if and only if $x > \frac{1}{2}$.

For example, the automaton depicted on Fig. 1 has value 1 if and only if $x > \frac{1}{2}$. The input alphabet is $A = \{a, b\}$, the initial state is the central state 0 and the final states are $F = \{L_3, R_2\}$. In order to maximize the probability to reach F, playing two b's in a row is certainly not a good option: from any of the three left states, this ensures to reach the non-accepting absorbing state L_2, thus any word in A^*bbA^* is accepted with probability less than $\frac{1}{2}$. A smarter strategy consists in playing long sequences of a's followed by one letter b. If $x \leq \frac{1}{2}$, there is still no hope to play a word accepted with probability greater than $\frac{1}{2}$, because playing letter a gives more chance to stay in L_1 than in R_1 thus playing ba^nb from state 0 gives more chance to reach the sink L_2 than the final state R_2. However, if $x > \frac{1}{2}$ then cleverly chosen

sequences $ba^{n_1}ba^{n_2}\cdots ba^{n_k}b$ are accepted with arbitrarily high probability (details can be found in [GO10], see also [BBG08] for a similar example).

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.

1.2.2 Informal description of the leaktight property

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" outside the class.

Such leaks occur in the automaton of Fig. 1 with the input sequence $(a^nb)_{n\in\mathbb{N}}$. As n grows large, the probability to reach L_2 and R_2 while reading the input word a^nb vanishes, thus the sets $\{L_1\}$ and $\{R_1\}$ are asymptotically recurrent. However there are leaks from L_1 to L_2 and symmetrically from R_1 to R_2. 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 the next section we prove that the value 1 problem is decidable when restricted to the subclass of leaktight automata.

1.2.3 Leaktight automata

The definition of a leaktight automaton relies on two key notions, idempotent words and word-recurrent states.

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

Definition 3 (Idempotent words)**.** *A finite word $u \in A^*$ is idempotent if for every states $s, t \in Q$,*

$$u(s,t) > 0 \iff (u \cdot u)(s,t) > 0 \ .$$

Idempotent words are everywhere: every word, if iterated a large number of times, becomes idempotent.

Lemma 1. *For every word $u \in A^*$, the word $u^{|Q|!}$ is idempotent.*

A finite word u induces naturally a Markov chain, which splits the set of states into two classes: recurrent states and transient states.

Definition 4 (Recurrent states)**.** *Let $u \in A^*$ be a finite word. A state s is* u-recurrent *if it is recurrent in the finite Markov chain $\mathcal{M}_u$ with states Q and transitions probabilities $(u(s,t))_{s,t\in Q}$.*

In the case of idempotent words, recurrence of a state can be easily characterized:

Lemma 2. *Let s be a state and u be an idempotent word. Then s is u-recurrent if and only if*

$$\forall t \in Q, u(s,t) > 0 \implies u(t,s) > 0 \ .$$

The formal definition of a leak is as follows:

Definition 5 (Leaks)**.** *A leak from a state $r \in Q$ to a state $q \in Q$ is a sequence $(u_n)_{n\in\mathbb{N}}$ of idempotent words such that:*

1. *for every $s, t \in Q$, the sequence $(u_n(s,t))_{n\in\mathbb{N}}$ converges to some value $u(s,t)$. We denote $\mathcal{M}_u$ the Markov chain with states Q and transition probabilities $(u(s,t))_{s,t\in Q}$,*

2. state r is recurrent in $\mathcal{M}_u$,

3. $\forall n \in \mathbb{N}, u_n(r,q) > 0$,

4. and r is not reachable from q in $\mathcal{M}_u$.

Condition 2. states that state r is asymptotically recurrent and conditions 3. and 4. express that some probability is "leaking" from outside the recurrence class of r without coming back.

Definition 6 (Leaktight automata)**.** *A probabilistic automaton is leaktight if it has no leak.*

Several examples of leaktight automata are given in Section 4.

2 The value 1 problem is decidable for leaktight automata

In this section we establish our main result:

Theorem 3. *The value* 1 *problem is decidable for leaktight automata.*

2.1 The Markov monoid algorithm

Our decision 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.

Algorithm 1 The Markov monoid algorithm.

Input: A probabilistic automaton with initial state i and final states F.
Output: Correct answer to the value 1 problem.

$\mathcal{G} \leftarrow \{\mathbf{a} \mid a \in A\}$.
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 $\mathcal{G}$
  if there is $\mathbf{u} \in \mathcal{G}$ such that $\mathbf{u} = \mathbf{u} \cdot \mathbf{u}$ and $\mathbf{u}^\sharp \notin \mathcal{G}$ **then**
    add $\mathbf{u}^\sharp$ to $\mathcal{G}$
until there is nothing to add
if there is a value 1 witness in $\mathcal{G}$ **then**
  return true
else
  return false

In the rest of the section, we explain the algorithm in details.

Definition 7 (limit-words)**.** *A* limit-word *is a map* $\mathbf{u} : Q^2 \to \{0,1\}$.

Initially, $\mathcal{G}$ only contains those limit-words $\mathbf{a}$ that are induced by input letters $a \in A$ of the automaton, where the limit-word $\mathbf{a}$ is defined by:

$$\forall s,t \in Q, (\mathbf{a}(s,t) = 1 \iff a(s,t) > 0) \ .$$

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

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 coincides with the multiplication of matrices with coefficients in the semiring $(\{0,1\}, \max, \min)$. The concatenation of two limit-words intuitively 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 $(u_n \cdot v_n)_{n\in\mathbb{N}}$.

Iteration of an idempotent limit-word. 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 an arbitrarily large function $f : \mathbb{N} \to \mathbb{N}$.

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

Definition 8 ($\mathbf{u}$-recurrence)**.** *A state s is $\mathbf{u}$-recurrent if for every state t,*

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

According to Lemma 2, this definition is consistent with the notion of recurrent state in Markov chains induced by finite words.

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

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

2.2 The Markov monoid and value 1 witnesses

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

Definition 9 (Markov monoid)**.** *The Markov monoid is the smallest set of limit-words containing $\{\mathbf{a} \mid a \in A\}$ 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.

On one hand, consistency ensures that every limit-word in $\mathcal{G}$ abstracts the asymptotic effect of an input sequence.

Definition 10 (Consistency)**.** *A set of limit-words $\mathcal{G} \subseteq \{0,1\}^{Q^2}$ is* consistent *with a probabilistic automaton $\mathcal{A}$ if for each limit-word $\mathbf{u} \in \mathcal{G}$, there exists a sequence of input words $(u_n)_{n\in\mathbb{N}}$ such that for every states $s, t \in Q$:*

$$\mathbf{u}(s,t) = 1 \iff \liminf_n u_n(s,t) > 0 \ . \tag{2}$$

Conversely, completeness ensures that every input sequence reifies one of the limit-words.

Definition 11 (Completeness)**.** *A set of limit-words $\mathcal{G} \subseteq \{0,1\}^{Q^2}$ is* complete *for a probabilistic automaton $\mathcal{A}$ if for each sequence of input words $(u_n)_{n\in\mathbb{N}}$, there exists $\mathbf{u} \in \mathcal{G}$ such that for every states $s, t \in Q$:*

$$\limsup_n u_n(s,t) = 0 \implies \mathbf{u}(s,t) = 0 \ . \tag{3}$$

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

Definition 12 (Value 1 witnesses)**.** *Let $\mathcal{A}$ be a probabilistic automaton. A* value 1 witness *is a limit-word* $\mathbf{u}$ *such that for every state* $s \in Q$*,*

$$\mathbf{u}(i,s) = 1 \implies s \in F \ . \tag{4}$$

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 4 (A criterion for value 1)**.** *Let $\mathcal{A}$ be a probabilistic automaton and $\mathcal{G} \subseteq \{0,1\}^{Q^2}$ be a set of limit-words. Suppose that $\mathcal{G}$ 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.*

In order to illustrate the interplay between limit-words of the Markov monoid and sequences of input words, we give a detailed proof of Lemma 4.

Proof. Assume first that $\mathcal{A}$ has value 1. Then for every $n \in \mathbb{N}$, there exists an input word $u_n \in A^*$ such that $\mathbb{P}_{\mathcal{A}}(u_n) \longrightarrow_n 1$. We know that $\sum_{f \in F} u_n(i,f) = \mathbb{P}_{\mathcal{A}}(u_n) \longrightarrow_n 1$. Since for all $n \in \mathbb{N}$, we have $\sum_{q \in Q} u_n(i,q) = 1$, then for all $s' \notin F$, $u_n(i,s') \longrightarrow_n 0$. Since $\mathcal{G}$ is complete, there exists a limit-word $\mathbf{u}$ such that (3) holds. Then $\mathbf{u}$ is a value 1 witness: let $s \in Q$ such that $\mathbf{u}(i,s) = 1$, then according to (3), $\limsup_n u_n(i,s) > 0$, hence $s \in F$.

Conversely, assume now that $\mathcal{G}$ contains a value 1 witness $\mathbf{u}$. Since $\mathcal{G}$ is consistent, there exists a sequence $(u_n)_{n \in \mathbb{N}}$ such that (2) holds. Without loss of generality, since $[0,1]^{Q \times Q}$ is compact, we can assume (by considering a convergent subsequence) that for every $s,t \in Q, u_n(s,t)$ converges to some limit $u(s,t)$. It follows from (2) and (4), for all $s \notin F$, we have $u_n(i,s) \longrightarrow_n 0$. Thus $\mathbb{P}_{\mathcal{A}}(u_n) = \sum_{f \in F} u_n(i,f) \longrightarrow_n 1$ *i.e* $\mathcal{A}$ has value 1. □

The following theorem proves that the Markov monoid of aleaktight automaton is consistent and complete, thus according to Lemma 4 it can be used to decide the value 1 problem.

Theorem 5. *The Markov monoid associated with an automaton $\mathcal{A}$ is consistent. Moreover if $\mathcal{A}$ is leaktight then the Markov monoid is complete.*

The proof of this theorem relies on a subtle algebraic argument based on the existence of factorization forests of bounded height [Sim90]. The same kind or argument was used by Simon to prove the decidability of the boundedness problem for distance automata [Sim94].

2.3 Correctness of the Markov monoid algorithm

Proposition 6. *The Markov monoid algorithm solves the value* 1 *problem for leaktight automata.*

In case 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.

Proposition 7. *If the Markov monoid algorithm outputs "true", the input probabilistic automaton has value* 1*.*

In case the Markov monoid algorithm outputs "false" and the automaton is leaktight then the value of the automaton can be bounded from above:

Theorem 8. *Let $\mathcal{A}$ be a probabilistic automaton whose minimal non-zero transition probability is denoted $p_{\min}$. If the Markov monoid algorithm outputs "false" and if moreover $\mathcal{A}$ is leaktight, then $val(\mathcal{A}) \leq 1 - p_{\min}^{3 \cdot 2^{4|Q|^2}}$.*

In the "false" case, one surely wishes to know whether the input automaton is leaktight or not. Fortunately, the leaktight property is decidable, as proved in the next section.

2.4 Complexity of the Markov monoid algorithm

Proposition 9. *The value* 1 *problem for leaktight automata is* PSPACE*-complete.*

The Markov monoid algorithm terminates in less than $2^{|Q|^2}$ iterations, since each iteration adds a new limit-word in the monoid and there are less than $2^{|Q|^2}$ different limit-words. This EXPTIME naïve upper bound can be improved to PSPACE.

A better complexity can be achieved by looking for the value 1 witness in a non-deterministic way. The algorithm guesses the value 1 witness and its decomposition by the product and iteration operations. The key observation made by Kirsten [Kir05] is that the $\sharp$-height (the number of nested applications of the iteration operation) can be restricted to at most $|Q|$. While Kirsten's proof was for desert automata, its adaptation to probabilistic automata can be roughly described as follows: idempotents in the same $\mathcal{J}$-class have the same number of connected components, and this quantity strictly decreases when iterating an unstable idempotent. The claim then follows from the remark that the number of connected components is bounded by $|Q|$, the number of states of the automaton.

Consequently, the value 1 problem can be decided in PSPACE. Conversely, there is a simple reduction from the emptiness problem for non-deterministic universal automata to the value 1 problem for leaktight automata: in fact, a non-deterministic universal automaton has value 1 if and only if there exists a word whose runs are all accepting. The first problem being PSPACE-hard [Koz77], the same applies to the value 1 problem for leaktight automata.

3 Deciding whether an automaton is leaktight

At first sight, the decidability of the leaktight property is not obvious: to check the existence of a leak one would need to scan the non-countable set of all possible sequences of input words. Still:

Theorem 10. *The leaktight property is decidable in polynomial space.*

Algorithm 2 The leak-finder algorithm.

Input: A probabilistic automaton with initial state i and final states F.
Output: Decides whether an automaton is leaktight.

$\mathcal{G}_+ \leftarrow \{(\mathbf{a}, \mathbf{a}) \mid a \in A\}$.
repeat
  if there is $(\mathbf{u}, \mathbf{u}_+), (\mathbf{v}, \mathbf{v}_+) \in \mathcal{G}_+$ such that $(\mathbf{u} \cdot \mathbf{u}, \mathbf{v}_+ \cdot \mathbf{v}_+) \notin \mathcal{G}_+$ **then**
    add $(\mathbf{u} \cdot \mathbf{v}, \mathbf{u}_+ \cdot \mathbf{v}_+)$ to $\mathcal{G}_+$
  if there is $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+$ such that $\mathbf{u} = \mathbf{u} \cdot \mathbf{u}$ and $\mathbf{u}_+ = \mathbf{u}_+ \cdot \mathbf{u}_+$ and $(\mathbf{u}^\sharp, \mathbf{u}_+) \notin \mathcal{G}_+$ **then**
    add $(\mathbf{u}^\sharp, \mathbf{u}_+)$ to $\mathcal{G}_+$
until there is nothing to add
if there is a leak witness in $\mathcal{G}_+$ **then**
  return false
else
  return true

The *leak-finder algorithm* deciding the leaktight property is very similar to the Markov monoid algorithm, except for two differences. First, the algorithm keeps track of those edges that are deleted by successive iteration operations. For that purpose, the algorithm stores together with each limit-word $\mathbf{u}$ another limit-word $\mathbf{u}_+$ to keep track of strictly positive transition probabilities. Such a pair $(\mathbf{u}, \mathbf{u}_+)$ is called an *extended limit-word* and the set of pairs of extended limit-words computed by the algorithm is called the *extended Markov monoid*. Second, the algorithm looks for *leak witnesses*.

Definition 13 (Leak witness)**.** *An extended limit-word* $(\mathbf{u}, \mathbf{u}_+)$ *is idempotent if both* $\mathbf{u}$ *and* $\mathbf{u}_+$ *are idempotent. An extended limit-word* $(\mathbf{u}, \mathbf{u}_+)$ *is a* leak witness *if it is idempotent and there exists* $r, q \in Q$ *such that:*

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

The correctness of the leak-finder algorithm is a consequence of:

Theorem 11. *An automaton* $\mathcal{A}$ *is leaktight if and only if its extended Markov monoid does not contain a leak witness.*

Although we chose to present Theorem 5 and Theorem 11 separately, their proofs are tightly linked.

4 A few leaktight automata

In this section, we present several properties and examples of leaktight automata.

4.1 Two basic examples

The automaton on Fig. 2 is leaktight. Its extended Markov monoid is depicted on the right-hand side. Each of the four directed graphs represents an extended limit-word $(\mathbf{u}, \mathbf{u}_+)$, the edges marked + are the edges that are in $\mathbf{u}_+$ but not in $\mathbf{u}$.

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 $a^\sharp$ and $b \cdot a^\sharp$.

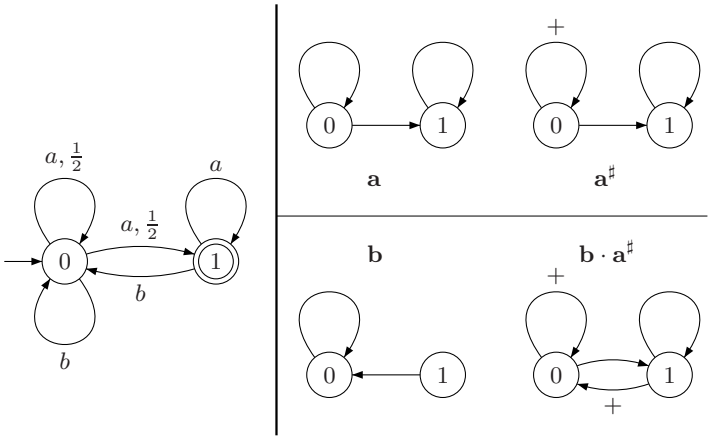

Figure 2: A leaktight automaton and its extended Markov monoid.

The automaton on Fig. 3 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.

4.2 The class of leaktight automata is rich and stable

The class of leaktight automata contains all known classes of probabilistic automata with a decidable value 1 problem, in particular hierarchical automata defined in [CSV09] and ♯-acyclic automata defined in [GO10].

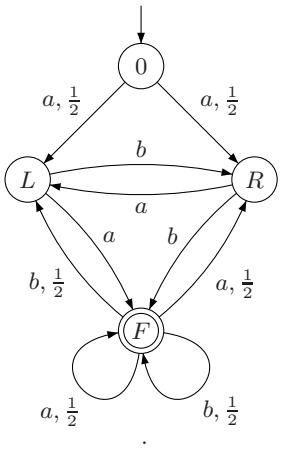

Figure 3: A leaktight automaton which does not have value 1.

Proposition 12. *Deterministic automata, hierarchical probabilistic automata and $\sharp$-acyclic automata are leaktight.*

Another witness of the interest of the class of leaktight automata is its stability under two natural composition operators: parallel composition and synchronized product. An automaton $\mathcal{A}||\mathcal{B}$ is the parallel composition of two automata $\mathcal{A}$ and $\mathcal{B}$ if its state space is the disjoint union of the state spaces of $\mathcal{A}$ and $\mathcal{B}$ plus a new initial state. For every input letter, the possible successors of the initial state are itself or one of the initial state of $\mathcal{A}$ and $\mathcal{B}$. An automaton $\mathcal{A} \times \mathcal{B}$ is the synchronized product of two automata $\mathcal{A}$ and $\mathcal{B}$ if its state space is the cartesian product of the state spaces of $\mathcal{A}$ and $\mathcal{B}$, with induced transition probabilities.

Proposition 13. *The leaktight property is stable by parallel composition and synchronized product.*

4.3 About $\sharp$-height

An adaptation of a result by Kirsten (Lemma 5.7 in [Kir05]) shows that only $|Q|$ nested applications of the iteration operation are sufficient to compute the whole Markov monoid of an automaton.

A natural question is whether this bound is tight? The answer is positive, a simple computation shows that the only value 1 witness of the automaton of Fig. 4 is $\mathbf{u} = (\cdots((\mathbf{a}_0^\sharp \mathbf{a}_1)^\sharp \mathbf{a}_2)^\sharp \mathbf{a}_3)^\sharp \cdots \mathbf{a}_{n-1})^\sharp$, whose $\sharp$-height is $n = |Q| - 2$.

The following proposition shows a crucial difference between $\sharp$-acyclic automata on one hand and deterministic and $\sharp$-acyclic automata on the other hand.

Proposition 14. *Deterministic automata and $\sharp$-acyclic automata have $\sharp$-height* 1.

As a consequence, every value 1 witness $\mathbf{u}$ can be rewritten: $\mathbf{u} = \mathbf{u}_0 \mathbf{v}_0^\sharp \mathbf{u}_1 \mathbf{v}_1^\sharp \cdots \mathbf{u}_k \mathbf{v}_k^\sharp \mathbf{u}_{k+1}$, where $\mathbf{u}_0, \mathbf{v}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{k+1}$ are obtained by concatenation of letters. By contrast, the value 1 witness of the automaton in Fig. 4 is more complex.

Acknowledgment

We thank Thomas Colcombet for having pointed us to the work of Leung and Simon.

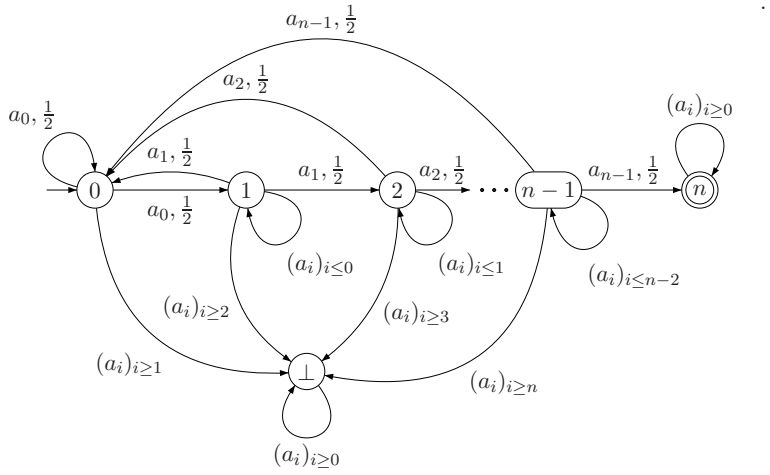

Figure 4: A leaktight automaton with value 1 and ♯-height n.

Conclusion

We introduced a subclass of probabilistic automata, called leaktight automata, for which we proved that the value 1 problem is PSPACE-complete.

In the present paper we considered automata over finite words. Next step is the adaptation of our results to infinite words and probabilistic Büchi automata [BBG08, CSV09], as well as partially observable Markov decision processes.

References

[BBG08] Christel Baier, Nathalie Bertrand, and Marcus Größer. On decision problems for probabilistic Büchi automata. In *Foundations Of Software Science And Computation Structures*, pages 287–301, 2008.

[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 of incomplete information. *Logical Methods in Computer Science*, 3(3), 2007.

[CK97] Karel Culik and Jarkko Kari. *Digital images and formal languages*, pages 599–616. Springer-Verlag New York, Inc., 1997.

[CL89] Anne Condon and Richard J. Lipton. On the complexity of space bounded interactive proofs (extended abstract). In *Foundations of Computer Science*, pages 462–467, 1989.

[CL04] Jérémie Chalopin and Hing Leung. On factorization forests of finite height. *Theoretical Computer Science*, 310(1-3):489–499, 2004.

[CMR07] Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. L_p distance and equivalence of probabilistic automata. *International Journal of Foundations of Computer Science*, 18(4):761–779, 2007.

[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.

[Col10] Thomas Colcombet. Factorization forests for infinite words and applications to countable scattered linear orderings. *Theoretical Computer Science*, 411(4-5):751–764, 2010.

[CSV09] Rohit Chadha, A. Prasad Sistla, and Mahesh Viswanathan. Power of randomization in automata on infinite strings. In *International Conference on Concurrency Theory*, pages 229–243, 2009.

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

[FGO11] Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Pushing undecidability of the isolation problem for probabilistic automata. *CoRR*, abs/1104.3054, 2011.

[GO10] Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *International Colloquium on Automata, Languages and Programming*, pages 527–538, 2010.

[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 proofs systems. In *Proceedings of 18th Symposium on the Foundations of Computer Science*, 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, 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. *Informatique Théorique et Applications*, 28(3-4):277–294, 1994.

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

Appendix

A Tool lemmata

We start with a few tool lemmata.

Lemma 1 *For every word $u \in A^*$, the word $u^{|Q|!}$ is idempotent. For every limit-word $\mathbf{u} \in \{0,1\}^{Q^2}$, the limit-word $\mathbf{u}^{|Q|!}$ is idempotent.*

Proof. The two statements are equivalent, we prove the second one.

Let $n = |Q|!$ and $s, t \in Q$ such that $\mathbf{u}^n(s,t) = 1$. We want to prove that $\mathbf{u}^{2n}(s,t) = 1$. Since $\mathbf{u}^n(s,t) = 1$, there exists $q \in Q$ and $k, l < |Q|$ such that $\mathbf{u}^k(s,q) = 1$, $\mathbf{u}^{n-k-l}(q,q) = 1$ and $\mathbf{u}^l(q,t) = 1$. Consequently, there exists $k' < |Q|$ such that $\mathbf{u}^{k'}(q,q) = 1$, and since $k' | n = |Q|!$, this implies $\mathbf{u}^n(q,q) = 1$, thus $\mathbf{u}^{2n-k-l}(q,q) = 1$ and finally $\mathbf{u}^{2n}(s,t) = 1$.

The proof of $\mathbf{u}^{2n}(s,t) = 1 \implies \mathbf{u}^n(s,t) = 1$ is similar. □

Lemma 2 *Let s be a state and u be an idempotent word. Then s is u-recurrent if and only if*

$$\forall t \in Q, u(s,t) > 0 \implies u(t,s) > 0 \ .$$

Proof. Suppose first that s is u-recurrent, and let $t \in Q$ such that $u(s,t) > 0$. Then t is accessible from s in $\mathcal{M}_u$ and by definition of recurrence, s is accessible from t in $\mathcal{M}_u$ thus there exists $k \in \mathbb{N}$ such that $u^k(t,s) > 0$. Since u is idempotent, an easy induction shows that $\exists l > 0, u^l(t,s) > 0 \implies u(t,s) > 0$.

Suppose now that for every $t \in Q$, $u(s,t) > 0 \implies u(t,s) > 0$. Let t be a state accessible from s, then there exists $k \in \mathbb{N}$ such that $u^k(s,t) > 0$. The same easy induction shows that $u(s,t) > 0$ thus by hypothesis $u(t,s) > 0$ and s is accessible from t in $\mathcal{M}_u$. Since this holds for every t accessible from s, the state s is recurrent in $\mathcal{M}_u$. □

The following lemma provides two simple yet useful properties.

Lemma 15. *Let $\mathcal{A}$ be a probabilistic automaton.*

i) Let $\mathbf{u}$ be an idempotent limit-word. Then for each state $s \in Q$, there exists $t \in Q$ such that $\mathbf{u}(s,t) = 1$ and t is $\mathbf{u}$-recurrent.

ii) Let $(\mathbf{v}, \mathbf{v}_+)$ be an element of the extended Markov monoid of $\mathcal{A}$. Then:

$$\forall s, t \in Q, (\mathbf{v}(s,t) = 1 \implies \mathbf{v}_+(s,t) = 1) \ . \tag{5}$$

Proof. We prove i). Let $C \subseteq Q$ be a strongly connected component of the graph $(Q, \mathbf{u})$ reachable from s. Then for every $t, q \in C$ there exists k and a path $t = t_1, t_2, \ldots, t_k = q$ from t to q in $(Q, \mathbf{u})$. Thus, $(\mathbf{u}^k)(t,q) = 1$ and since $\mathbf{u}$ is idempotent, $\mathbf{u}(t,q) = 1$. Thus, C is a clique of the graph $(Q, \mathbf{u})$ and all states of C are recurrent. Since C is reachable from s in $(Q, \mathbf{u})$ then the same argument proves that for every $t \in C$, $\mathbf{u}(s,t) = 1$.

We prove ii). By definition, the extended Markov monoid is the smallest monoid containing $\{(\mathbf{a}, \mathbf{a}) \mid a \in A\}$ and stable by concatenation and iteration. Property (5) holds for every pair $(\mathbf{a}, \mathbf{a})$ where $a \in A$. Moreover this property is stable by concatenation and iteration. This completes the proof. □

B Correctness of the Markov monoid algorithm

Theorem 5 *The Markov monoid associated with an automaton $\mathcal{A}$ is consistent. Moreover if $\mathcal{A}$ is leaktight then the Markov monoid is complete.*

Proof. The proof is split between Lemma 16 and Lemma 17. □

B.1 The input sequence associated with a limit-word

Definition 14. *A sequence $(u_n)_{n\in\mathbb{N}}$ of input words reifies a limit-word $\mathbf{u}$ if*

$$\mathbf{u}(s,t) = 1 \iff \liminf_n u_n(s,t) > 0 \ . \tag{6}$$

In particular, a set of limit-words $\mathcal{G}$ is consistent for $\mathcal{A}$ if each limit-word in $\mathcal{G}$ is reified by some sequence of input words.

Lemma 16. *Let $\mathcal{G} \subseteq \{0,1\}^{Q^2}$ be a set of limit-words. Suppose that $\mathcal{G}$ is consistent. Then for every $\mathbf{u}, \mathbf{v} \in \mathcal{G}$ the set $\mathcal{G} \cup \{\mathbf{u}\cdot\mathbf{v}\}$ is consistent. If moreover $\mathbf{u}$ is idempotent then $\mathcal{G} \cup \{\mathbf{u}^\sharp\}$ is consistent as well.*

Proof. Let $\mathbf{u}, \mathbf{v} \in \mathcal{G}$.

We build a sequence $(w_n)_{n\in\mathbb{N}}$ which reifies $\mathbf{u}\cdot\mathbf{v}$. By induction hypothesis on $\mathbf{u}$ and $\mathbf{v}$, there exists $(u_n)_n$ and $(v_n)_n$ which reify $\mathbf{u}$ and $\mathbf{v}$ respectively. Let $w_n = u_n \cdot v_n$. Then $(w_n)_{n\in\mathbb{N}}$ reifies $\mathbf{u}\cdot\mathbf{v}$, because

$$w_n(s,t) = \sum_{r\in Q} u_n(s,t)\cdot v_n(t,r)$$

and by definition of the concatenation of two limit-words.

Suppose now that $\mathbf{u}$ is idempotent, we build a sequence $(z_n)_{n\in\mathbb{N}}$ which reifies $\mathbf{u}^\sharp$. By induction hypothesis, there exists a sequence $(u_n)_{n\in\mathbb{N}}$ which reifies $\mathbf{u}$. Since $[0,1]^{Q\times Q}$ is compact, we can suppose (by considering subsequences if needed) that for every $s,t\in Q$, $u_n(s,t)$ converges to some value $u(s,t)$. Since $\mathbf{u}$ is idempotent then u is idempotent as well. As a consequence, the Markov chain $\mathcal{M}_u$ with state space Q and transition probabilities $(u(s,t))_{s,t\in Q}$ is aperiodic, and according to standard results about finite Markov chains, the sequence of matrices $(u^k)_{k\in\mathbb{N}}$ has a limit $z \in [0,1]^{Q\times Q}$ such that transient states of z are asymptotically left forever. This implies:

$$\forall s,t\in Q, z(s,t) > 0 \implies t \text{ is } z\text{-recurrent.} \tag{7}$$

Since $(u_n)_{n\in\mathbb{N}}$ converges to u and since matrix product is continuous, for every $k\in\mathbb{N}$ there exists $\phi(k)\in\mathbb{N}$ such that $||u^k - u^k_{\phi(k)}||_\infty \leq \frac{1}{k}$. Then the sequence of matrices $z_n = u^n_{\phi(n)}$ converges to z.

Now we prove that $(z_n)_{n\in\mathbb{N}}$ reifies $\mathbf{u}^\sharp$ because,

$$\begin{aligned}
\mathbf{u}^\sharp(s,t) = 1 &\iff t \text{ is } \mathbf{u}\text{-recurrent and } \mathbf{u}(s,t) = 1\\
&\iff t \text{ is } u\text{-recurrent and } u(s,t) > 0\\
&\iff t \text{ is } z\text{-recurrent and } z(s,t) > 0\\
&\iff z(s,t) > 0\\
&\iff \lim_n z_n(s,t) > 0 \ ,
\end{aligned}$$

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 because the iterated Markov chain $z = \lim_k u^k$ has the same recurrent states than the Markov chain u, the fourth holds by (7), and the fifth by definition of z. □

Proposition 9 *The Markov monoid algorithm solves the value* 1 *problem for leaktight automata.*

Proof. Termination of the Markov monoid algorithm is straightforward because each iteration adds a new element in $\mathcal{G}$ and there are at most $2^{|Q|^2}$ elements in $\mathcal{G}$.

The correctness is a corollary of Theorem 5: since the Markov monoid is consistent and complete then according to Lemma 4, $\mathcal{A}$ has value 1 if and only if $\mathcal{G}$ contains a value 1 witness, if and only if the Markov monoid algorithm outputs "true". □

Proposition 7 *If the Markov monoid algorithm outputs "true", the input probabilistic automaton has value* 1.

Proof. According to Theorem 5, the Markov monoid is consistent. If it contains a value 1 witness, then according to the first part of the proof of Lemma 4, $\mathcal{A}$ has value 1. □

Theorem 8 *If the Markov monoid algorithm outputs "false" and if moreover the input probabilistic automaton $\mathcal{A}$ is leaktight, then val$(\mathcal{A}) \leq 1 - p_{\min}^{3 \cdot 2^{4|Q|^2}}$.*

Proof. If the algorithm outputs "false", then for every pair $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+$, the limit-word $\mathbf{u}$ is not a value 1 witness, because by definition of the Markov monoid $\mathcal{G}$ and the extended Markov monoid $\mathcal{G}_+$,

$$\mathcal{G} = \{\mathbf{u} \mid \exists \mathbf{u}_+, (\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+\} \ .$$

Let $u \in A^*$ be any finite word. According to the Lower Bound Lemma (Lemma 18), there exists a pair $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+$ such that for every $s, t \in Q$,

$$\mathbf{u}(s,t) = 1 \implies u(s,t) \geq p_{\min}^{3 \cdot 2^{4|Q|^2}} \ . \tag{8}$$

Since $\mathbf{u}$ is not a value 1 witness, there exists $q \notin F$ such that $\mathbf{u}(i,q) = 1$ then according to (8), $u(i,q) \geq p_{\min}^{3 \cdot 2^{4|Q|^2}}$, thus $\sum_{f \in F} u(i,f) \leq 1 - p_{\min}^{3 \cdot 2^{4|Q|^2}}$.

Since this holds for every $u \in A^*$, $\text{val}(\mathcal{A}) \leq 1 - p_{\min}^{3 \cdot 2^{4|Q|^2}}$. □

B.2 The limit-word associated with an input sequence

In this section, we show how to associate a limit-word with each infinite sequence of input words.

Our goal is to prove:

Lemma 17. *Suppose that $\mathcal{A}$ is leaktight. Then the Markov monoid of $\mathcal{A}$ is complete for $\mathcal{A}$.*

The proof relies on two lemmata, Lemma 23 and the Lower Bound Lemma (Lemma 18), which are proved in the next section.

Proof of Lemma 17. Let $(u_n)_{n \in \mathbb{N}}$ be an input sequence.

According to Lemma 23, the extended Markov monoid of $\mathcal{A}$ contains no leak witness, thus we can apply the Lower Bound Lemma to each input word u_n: for each $n \in \mathbb{N}$ there exists a pair $(\mathbf{u}_n, \mathbf{u}_{+,n}) \in \mathcal{G}_+$ in the extended Markov monoid such that:

$$\begin{aligned} \mathbf{u}_{+,n}(s,t) = 1 &\iff u_n(s,t) > 0 \ , \\ \mathbf{u}_n(s,t) = 1 &\implies u_n(s,t) \geq p_{\min}^{3 \cdot 2^{4|Q|^2}} \ . \end{aligned} \tag{9}$$

The set of limit-words is finite, thus 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 $\mathbf{u}_N$ has the property (3) defining completeness:

$$\forall s, t \in Q, \limsup u_n(s,t) = 0 \implies \mathbf{u}_N(s,t) = 0 \ . \tag{10}$$

If $\limsup u_n(s,t) = 0$ then $u_n(s,t) < p_{\min}^{3 \cdot 2^{4|Q|^2}}$ for every n sufficiently large. Since $\mathbf{u}_N = \mathbf{u}_n$ for infinitely many $n \in \mathbb{N}$, then according to (9) this implies $\mathbf{u}_N(s,t) = 0$, which completes the proof of Lemma 17. □

C The Lower Bound Lemma

The Lower Bound Lemma is the key to our decidability result.

Lemma 18 (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 every word $u \in A^*$, there exists a pair $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid such that, for all states $s, t \in Q$,*

$$\mathbf{u}_+(s,t) = 1 \iff u(s,t) > 0 \ , \tag{11}$$

$$\mathbf{u}(s,t) = 1 \implies u(s,t) \geq p_{\min}^{3 \cdot 2^{4|Q|^2}} \ . \tag{12}$$

To prove Lemma 18, we rely on the notion of Ramseyan factorization trees and decomposition trees introduced by Simon [Sim90, Sim94].

Definition 15. *Let A be a finite alphabet, $(M, \cdot, 1)$ a monoid and $\phi : A^* \to M$ a morphism. A* Ramseyan factorization tree *of a word $u \in A^*$ for ϕ is a finite unranked ordered tree, whose nodes are labelled by pairs $(w, \phi(w))$ where w is a word in A^* and such that:*

(i) the root is labelled by $(u, \phi(u))$,

(ii) every internal node with two children labelled by $(u_1, \phi(u_1))$ and $(u_2, \phi(u_2))$ is labelled by $(u_1 \cdot u_2, \phi(u_1 \cdot u_2))$,

(iii) leaves are labelled by pairs $(a, \phi(a))$ with $a \in A$,

(iv) if an internal node t has n children $t_1, \ldots, t_n$ where n is greater or equal to three, labelled by $(u_1, \phi(u_1)), \ldots, (u_n, \phi(u_n))$, then there exists $\mathbf{e} \in M$ such that $\mathbf{e}$ is idempotent and $\mathbf{e} = \phi(u_1) = \phi(u_2) = \ldots = \phi(u_n)$. In this case t is labelled by $(u_1 \cdots u_n, \mathbf{e})$.

Internal nodes with one or two children are concatenation nodes*, the other internal nodes are* iteration nodes*.*

Not surprisingly, every word $u \in A^*$ can be factorized as a Ramseyan factorization tree, using only concatenation nodes: any *binary* tree whose leaves are labelled from left to right by the letters of u and whose internal nodes are labelled consistently is a Ramseyan factorization tree. Notice that if u has length n then such a tree has height $\log_2(n)$, with the convention that the height of a leaf is 0. As a consequence, with this naive factorization of u, the longer the word u, the deeper its factorization tree.

The following powerful result of Simon states that every word can be factorized with a Ramseyan factorization tree whose depth is bounded independently of the length of the word:

Theorem 19 ([Sim90, CL04, Col10])**.** *Every word $u \in A^*$ has a Ramseyan factorization tree of height at most $3 \cdot |M|$.*

In [Sim94], Simon used the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +)$ to prove the decidability of the boundedness problem for distance automata. Similarly to the Markov monoid, the tropical semiring is equipped with an iteration operation $\sharp$. Following the proof scheme of Simon, we introduce the notion of decomposition tree relatively to a monoid M equipped with an iteration operation $\sharp$.

Definition 16. *Let A be a finite alphabet, $(M, \cdot, 1)$ a monoid equipped with an iteration operation $\sharp$ that maps every idempotent $\mathbf{e} \in M$ to another idempotent element $\mathbf{e}^\sharp \in M$ and $\phi : A^* \to M$ a morphism. A* decomposition tree *of a word $u \in A^*$ is a finite unranked ordered tree, whose nodes have labels in (A^*, 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 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 leaf is labelled by $(a, \mathbf{a})$ *where* a *is a letter,*

iv) for every internal node with three or more children, there exists $\mathbf{e} \in M$ *such that* $\mathbf{e}$ *is idempotent and the node is labelled by* $(u_1 \dots u_n, \mathbf{e}^\sharp)$ *and its children are labelled by* $(u_1, \mathbf{e}), \dots, (u_n, \mathbf{e})$*.*

Internal nodes with one or two children are concatenation *nodes, the other internal nodes are* iteration *nodes.*

An iteration node labelled by $(u, \mathbf{e})$ *is* discontinuous *if* $\mathbf{e}^\sharp \neq \mathbf{e}$*. The* span *of a decomposition tree is the maximal length of a path that contains no discontinuous path.*

Remark that decomposition and factorization trees are closely related:

Lemma 20. *A Ramseyan factorization tree is a decomposition tree if and only if it contains no discontinuous nodes.*

Proof. The definitions 15 and 16 are similar except for condition iv). If there are no discontinuous nodes then $\mathbf{e} = \mathbf{e}^\sharp$ in iv) of Definition 16. □

The following theorem is adapted from [Sim94, Lemma 10].

Theorem 21. *Let* A *be a finite alphabet,* $(M, \cdot, 1)$ *a monoid equipped with an iteration operation* $\sharp$ *that maps every idempotent* $\mathbf{e} \in M$ *to another idempotent element* $\mathbf{e}^\sharp \in M$ *and* $\phi : A^* \to M$ *a morphism. Every word* $u \in A^*$ *has a decomposition tree whose span is less than* $3 \cdot |M|$*.*

Proof. Let K be the cardinal of M. We start with adding a few letters to A, together with their transition matrices. For every idempotent $\mathbf{e} \in M$, we add a letter $\underline{e}$ to the alphabet A, and extend ϕ by $\phi(\underline{\mathbf{e}}) = \mathbf{e}$. We do not lose generality because this operation does not modify the semigroup M, and a decomposition tree for a word with letters from the original alphabet cannot use the new letters of the extended alphabet $\underline{A} = A \cup \{\underline{\mathbf{e}} \mid \mathbf{e} \in M, \mathbf{e} = \mathbf{e}^2\}$.

We proceed by induction on the length of u.

First, if u is a letter a, the decomposition tree whose only node is labelled by $(a, \phi(a))$ has span 1.

Consider now a word u with at least two letters. According to Theorem 19, there exists a Ramseyan factorization tree T_u of u of height less than $3 \cdot K$. According to Lemma 20, this factorization tree is in general not a decomposition tree, except in the very special case where it has no discontinuous nodes. The rest of the proof shows how to transform T_u into a decomposition tree of span less than $3 \cdot |M|$ in the following way. The proof technique consists in taking care of the discontinuous nodes of T_u in a bottom-up fashion.

If T_u has no discontinuous node then it is already a decomposition tree. Moreover its span is equal to its height, which is less than $3 \cdot |M|$.

If T_u has at least one discontinuous node, and let t be such a node with maximal depth. By definition, t has at least three children $t_1, \dots, t_k$ labelled by $(v_1, \phi(v_1)), \dots, (v_k, \phi(v_k))$ and t itself is labelled by $(v, \phi(v)) = (v_1 \cdots v_k, \phi(v_1 \cdots v_k))$. We have $\phi(v_1) = \dots = \phi(v_k) = e$ and since t is discontinuous, $e \neq e^\sharp$. We distinguish two cases.

- Either t is the root of T_u. In that case $v = u$ and we can construct directly a decomposition tree T_u of u of span less than $3 \cdot |M|$. Let $T_1, \dots, T_k$ the subtrees of T_u whose roots are respectively $t_1, \dots, t_k$. Then, since t is a discontinuous node of maximal depth in T_u, each subtree $T_1, \dots, T_k$ contains no discontinuous node at all. Consequently, each subtree T_i

is a decomposition tree whose span is equal to its height, which is less than $3 \cdot |M| - 1$. Then a decomposition tree for u is the tree with the root labelled by $(u, e^\sharp)$ and children $T_1, \ldots, T_k$. Since $e \neq e^\sharp$, the root is discontinuous and the span of this tree is less than $3 \cdot |M|$.

- Or t is not the root of T_u. Since t is labelled by $(v, e^\sharp)$, there exist two words $w, w' \in A^*$ such that $u = w \cdot v \cdot w'$. Let replace the subword v in u by the letter $\underline{\mathbf{e}^\sharp}$ and obtain the word $u' = w \cdot \underline{\mathbf{e}^\sharp} \cdot w'$. Since t is a discontinuous node, it is an iteration node and has at least three children, thus v has length at least 3. Thus, u' is strictly shorter than u and we can apply the induction hypothesis to u': let T' be a decomposition tree for u', whose span is less than $3 \cdot |M|$. One of the leaf of T' corresponds to the letter $\underline{\mathbf{e}^\sharp}$ of u' and is labelled by $(\underline{\mathbf{e}^\sharp}, \mathbf{e}^\sharp)$. Let replace this leaf by the decomposition tree T_v of v given by induction hypothesis. Since T_v is labelled by $(v, \mathbf{e}^\sharp)$, we obtain a decomposition tree for $u = w \cdot v \cdot w'$, whose span is less than $3 \cdot |M|$. This completes the induction step.

□

The following lemma proves that every discontinuous node of a decomposition tree strictly decreases the $\mathcal{J}$-class of the second label of the node. The notion of $\mathcal{J}$-class of a monoid M is a classical notion in semigroup theory, derived from one of the four Green's relations (for details about Green's relations, see [Lal79, How95]). The $\mathcal{J}$-preorder between elements of a monoid M is defined as follows:

$$\forall a, b \in M, a \leq_{\mathcal{J}} b \text{ if } a \in MbM \ ,$$

where MbM denotes the set $\{ubv \mid u, v \in M\}$.

The following lemma is adapted from [Sim94, Lemma 3].

Lemma 22. *Let A be a finite alphabet, and M a monoid equipped with an iteration operation $\sharp$ that maps every idempotent $e \in M$ to another idempotent element $e^\sharp \in M$. Suppose that for every idempotent $e \in M$,*

$$e^\sharp \cdot e = e^\sharp = e \cdot e^\sharp \ . \tag{13}$$

Then for every idempotent element $e \in M$, either $e^\sharp = e$ or $e^\sharp <_{\mathcal{J}} e$.

Proof. Equation (13) implies that $e^\sharp = e^\sharp e e^\sharp$ thus $e^\sharp \leq_{\mathcal{J}} e$. Now, we suppose that $e \leq_{\mathcal{J}} e^\sharp$ and prove that $e = e^\sharp$. Since M is finite, we have $e \mathcal{D} e^\sharp$. Since $e \cdot e^\sharp = e^\sharp$, it follows $e \mathcal{R} e^\sharp$. By a dual argument, we have $e \mathcal{L} e^\sharp$; hence $e \mathcal{H} e^\sharp$. Both e and $e^\sharp$ and idempotents, so according to Green's theorem (see e.g. [How95], Theorem 2.2.5.) $e = e^\sharp$. □

Now we are ready to complete the proof of the Lower Bound lemma.

Proof of Lemma 18. Let M be the extended Markov monoid $\mathcal{G}_+$ associated with $\mathcal{A}$ and equipped with the concatenation operation:

$$(\mathbf{u}, \mathbf{u}_+) \cdot (\mathbf{v}, \mathbf{v}_+) = (\mathbf{u} \cdot \mathbf{v}, \mathbf{u}_+ \cdot \mathbf{v}_+) \ ,$$

and for idempotent pairs the iteration operation:

$$(\mathbf{u}, \mathbf{u}_+)^\sharp = (\mathbf{u}^\sharp, \mathbf{u}_+) \ .$$

We apply Theorem 21 to the word u, the extended Markov monoid $M = \mathcal{G}_+$ and the morphism $\phi : A \to M$ defined by $\phi(a) = (\mathbf{a}, \mathbf{a})$. Let $u \in A^*$, according to Theorem 21, u has a decomposition tree T of span less than $3 \cdot |\mathcal{G}_+|$, labelled by $(u, \mathbf{u})$ for some limit-word $\mathbf{u}$.

We prove that:

$$\text{the depth of } T \text{ is less than } 3 \cdot |\mathcal{G}_+|^2\text{-1}. \tag{14}$$

Let $t_0, t_1, \dots t_n$ be a path from the root to a leaf in T and for each i denote $(u_i, \mathbf{u}_i)$ the label of t_i (in particular, $(u_0, \mathbf{u}_0) = (u, \mathbf{u})$). Then, for every $k \in \mathbb{N}$ such that t_k is a discontinuous node, by Lemma 22 $\mathbf{u}_k <_{\mathcal{J}} \mathbf{u}_{k+1}$. Moreover, for every $k \in \mathbb{N}$ such that t_k is a continuous node, by definition of a decomposition tree, $\mathbf{u}_k \leq_{\mathcal{J}} \mathbf{u}_{k+1}$. Since the span is less than $3 \cdot |\mathcal{G}_+|$, and since there are less $\mathcal{J}$-classes than there are elements in the monoid $\mathcal{G}_+$, this gives (14).

To complete the proof of Lemma 18, we prove that for every word $u \in A^*$ with a decomposition tree of depth h, there exists a pair $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid such that, for all states $s, t \in Q$,

$$\mathbf{u}_+(s,t) = 1 \iff u(s,t) > 0 \ , \tag{15}$$

$$\mathbf{u}(s,t) = 1 \implies u(s,t) \geq p_{\min}^{2^{h+1}} \ . \tag{16}$$

We prove (15) and (16) by induction on h.

If $h = 0$ then the decomposition tree is a leaf, hence u is a letter a and $\mathbf{u} = \mathbf{u}_+ = \mathbf{a}$. Then (15) holds by definition of $\mathbf{a}$ and (16) holds by definition of $p_{\min}$.

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

First case, t is a concatenation node labelled by $(u, (\mathbf{u}, \mathbf{u}_+))$ with two sons labelled by $(u_1, (\mathbf{u}_1, \mathbf{u}_{+,1}))$ and $(u_2, (\mathbf{u}_2, \mathbf{u}_{+,2}))$. We first prove that (15) holds. Let $s, t \in Q$ such that $\mathbf{u}_+(s,t) = 1$. By definition of a decomposition tree, $\mathbf{u}_+ = \mathbf{u}_{+,1} \cdot \mathbf{u}_{+,2}$. Since $\mathbf{u}_+(s,t) = 1$ then by definition of the concatenation there exists $q \in Q$ such that $\mathbf{u}_{+,1}(s,q) = 1$ and $\mathbf{u}_{+,2}(q,t) = 1$. Then $u(s,t) \geq u_1(s,q) \cdot u_2(q,t)$ and by induction hypothesis $u_1(s,q) \cdot u_2(q,t) > 0$, which proves (15). Now we prove that (16) holds. Let $s, t \in Q$ such that $\mathbf{u}(s,t) = 1$. By definition of a decomposition tree, $\mathbf{u} = \mathbf{u}_1 \cdot \mathbf{u}_2$. Since $\mathbf{u}(s,t) = 1$ then by definition of limit-words concatenation there exists $q \in Q$ such that $\mathbf{u}_1(s,q) = 1$ and $\mathbf{u}_2(q,t) = 1$. Then $u(s,t) \geq u_1(s,q) \cdot u_2(q,t) \geq p_{\min}^{2^h} \cdot p_{\min}^{2^h} = p_{\min}^{2^{h+1}}$ where the first inequality is by definition of the matrix product and the second inequality is by induction hypothesis. This completes the proof of (16).

Second case, t is an iteration node labelled by $(u, (\mathbf{u}^\sharp, \mathbf{u}_+))$ with k sons labelled by $(u_1, (\mathbf{u}, \mathbf{u}_+)), \dots, (u_k, (\mathbf{u}, \mathbf{u}_+))$. The proof that (15) holds is similar to the concatenation node case. Now we prove that (16) holds.

Let $s, r \in Q$ such that $\mathbf{u}^\sharp(s,r) = 1$. By definition of a decomposition tree, $\mathbf{u} = \mathbf{u}_1 \cdots \mathbf{u}_k$. Since t is an iteration node, $k \geq 3$ thus:

$$u(s,t) \geq u_1(s,r) \cdot \sum_{q \in Q} (u_2 \cdots u_{k-1})(r,q) \cdot u_k(q,t) \ . \tag{17}$$

To establish (16) we prove that:

$$u_1(s,r) \geq p_{\min}^{2^h}, \tag{18}$$

$$\forall q \in Q, (u_2 \cdots u_{k-1})(r,q) > 0 \implies u_k(q,t) \geq p_{\min}^{2^h}. \tag{19}$$

First we prove (18). Since $\mathbf{u}^\sharp(s,r) = 1$ then by definition of the iteration operator, r is $\mathbf{u}$-recurrent and $\mathbf{u}(s,r) = 1$. By induction hypothesis applied to t_1, according to (16), it implies $u_1(s,r) \geq p_{\min}^{2^h}$ *i.e* (18).

Now we prove (19). For that we use the hypothesis that $(\mathbf{u}, \mathbf{u}_+)$ is not a leak witness. Let $q \in Q$ such that $(u_2 \cdots u_{k-1})(r,q) > 0$. Then by induction hypothesis applied to $t_2, \dots, t_{k-1}$, according to (15), $\mathbf{u}_+^{k-2}(r,q) = 1$. Thus by idempotence of $\mathbf{u}_+$, $\mathbf{u}_+(r,q) = 1$. Since r is $\mathbf{u}$-recurrent and since $(\mathbf{u}, \mathbf{u}_+)$ is not a leak witness then necessarily $\mathbf{u}(q,r) = 1$. Thus, by induction hypothesis and according to (16), $u_k(q,t) \geq p_{\min}^{2^h}$ *i.e* (19).

Now, putting (17), (18) and (19) altogether,

$$\begin{aligned} u(s,t) &\geq u_1(s,r) \sum_{q\in Q} (u_2 \cdots u_{k-1})(r,q) \cdot u_k(q,t) \\ &\geq p_{\min}^{2^h} \sum_{q\in Q} (u_2 \cdots u_{k-1})(r,q) \cdot p_{\min}^{2^h} \\ &\geq p_{\min}^{2^{h+1}}, \end{aligned}$$

where the second inequality holds because $\sum_{q\in Q}(u_2 \cdots u_{k-1})(r,q) = 1$. This completes the proof of (16).

To conclude, according to (14) the depth of a decomposition tree can be bounded by $3 \cdot |\mathcal{G}_+|^2$, and since $\mathcal{G}_+$ has less than $2^{2|Q|^2}$ elements the depth h is less than $3 \cdot 2^{4|Q|^2}$ Then according to (15) and (16) this completes the proof of Lemma 18. □

D About the extended Markov monoid and leak witnesses

Theorem 11 *An automaton $\mathcal{A}$ is leaktight if and only if its extended Markov monoid contains no leak witness.*

The proof is split in two parts, the direct implication (Lemma 23) and the converse implication (Lemma 24).

Lemma 23. *If the extended Markov monoid of an automaton $\mathcal{A}$ contains a leak witness then $\mathcal{A}$ has a leak.*

Proof. Suppose that there is a leak witness $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid.

By definition of a leak witness, $\mathbf{u}$ and $\mathbf{u}_+$ are idempotent and there exists $r, q \in Q$ such that r is $\mathbf{u}$-recurrent, $\mathbf{u}_+(r,q) = 1$ and $\mathbf{u}(q,r) = 0$.

We prove now that there exists a leak from r to q.

By induction, the proof of Lemma 16 proves that for each $(\mathbf{u}, \mathbf{u}_+)$ in the extended Markov monoid, there exists a sequence $(u_n)_{n\in\mathbb{N}}$ such that $\forall s,t \in Q$,

$$\mathbf{u}(s,t) = 1 \iff \liminf_n u_n(s,t) > 0 \ , \tag{20}$$

$$\mathbf{u}_+(s,t) = 0 \iff \forall n \in \mathbb{N}, u_n(s,t) = 0 \ , \tag{21}$$

$$\mathbf{u}_+(s,t) = 1 \iff \forall n \in \mathbb{N}, u_n(s,t) > 0 \ . \tag{22}$$

Since the set $[0,1]^{Q\times Q}$ is compact, we can extract from $(u_n)_{n\in\mathbb{N}}$ a subsequence $(u'_n)_{n\in\mathbb{N}}$ such that for every $s,t \in Q$, $(u'_n(s,t))_{n\in\mathbb{N}}$ converges to some limit $u(s,t)$.

To complete the proof, we show that $(u'_n)_{n\in\mathbb{N}}$ is a leak in $\mathcal{A}$ from r to q. According to Definition 5, there are four conditions to be met.

First, convergence of $(u'_n)_{n\in\mathbb{N}}$ is by choice of $(u'_n)_{n\in\mathbb{N}}$. Moreover, since $\mathbf{u}_+$ is idempotent then according to (20) and (21) all the u'_n are idempotent. Second, let $\mathcal{M}_u$ the Markov chain associated with transition probabilities $(u(s,t))_{s,t\in Q}$. We prove that r is $\mathcal{M}_u$-recurrent. Since r is $\mathbf{u}$-recurrent, and according to (20), $\forall s \in Q, u(r,s) > 0 \implies u(s,r) > 0$. But u is idempotent because $\mathbf{u}$ is idempotent and (20). Thus according to Lemma 2, r is u-recurrent. Third, $\forall n \in \mathbb{N}, u'_n(r,q) > 0$, this holds because of (22) and $\mathbf{u}_+(r,q) = 1$. Fourth, r is not accessible from q in $\mathcal{M}_u$. Since $u = \lim_n u'_n$ and according to (20), $\mathbf{u}(s,t) = 1 \iff u(s,t) > 0$, thus accessibility in the directed graph $(Q, \mathbf{u})$ and in the Markov chain $\mathcal{M}_u$ coincide. Since $\mathbf{u}(r,q) = 0$ and $\mathbf{u}$ is idempotent, then r is not accessible from q in $\mathbf{u}$, thus neither accessible in $\mathcal{M}_u$. □

Lemma 24. *If the extended Markov monoid of an automaton $\mathcal{A}$ contains no leak witness then $\mathcal{A}$ is leaktight.*

Proof. By contradiction, suppose there is a leak $(u_n)_{n\in\mathbb{N}}$ from a state r to a state q in $\mathcal{A}$, and for each $s,t \in q$, denote $u(s,t)$ the limit of the sequence $(u_n(s,t))_{n\in\mathbb{N}}$ and $\mathcal{M}_u$ the Markov chain induced by $(u(s,t))_{s,t\in Q}$. By definition of a leak:

$$r \text{ is recurrent in } \mathcal{M}_u, \tag{23}$$
$$\forall n \in \mathbb{N}, u_n(r,q) > 0, \tag{24}$$
$$r \text{ is not accessible from } q \text{ in } \mathcal{M}_u. \tag{25}$$

To get the contradiction, we use the leak $(u_n)_{n\in\mathbb{N}}$ to build a leak witness $(\mathbf{v},\mathbf{v}_+)$ in the extended monoid $\mathcal{G}_+$ of $\mathcal{A}$.

The first task is to define the pair $(\mathbf{v},\mathbf{v}_+)$. By hypothesis, we can apply the Lemma 18 to each word u_n of the leak, which gives for each $n \in \mathbb{N}$ a pair $(\mathbf{u}_n,\mathbf{u}_{+,n}) \in \mathcal{G}_+$ such that:

$$\mathbf{u}_{+,n}(s,t) = 1 \iff u_n(s,t) > 0 \ , \tag{26}$$
$$\mathbf{u}_n(s,t) = 1 \implies u_n(s,t) \geq p_{\min}^{3\cdot 2^{4|Q|^2}} \ . \tag{27}$$

Since the extended Markov monoid is finite, there exists $N \in \mathbb{N}$ such that:

$$\text{for infinitely many } n \in \mathbb{N}, (\mathbf{u}_N,\mathbf{u}_{+,N}) = (\mathbf{u}_n,\mathbf{u}_{+,n}) \ . \tag{28}$$

Let $(\mathbf{v},\mathbf{v}_+) = (\mathbf{u}_N,\mathbf{u}_{+,N})^{|\mathcal{G}_+|!}$. Then according to Lemma 1, $(\mathbf{v},\mathbf{v}_+)$ is idempotent. Note also that according to (26) and since the u_n are idempotent (by definition of leaks),

$$\mathbf{u}_{+,N} \text{ is idempotent and } \mathbf{v}_+ = \mathbf{u}_{+,N} \ . \tag{29}$$

Now, we prove that $(\mathbf{v},\mathbf{v}_+)$ is a leak witness. According to i) of Lemma 15, since $\mathbf{v}$ is idempotent, there exists r' such that $\mathbf{v}(r,r') = 1$ and r' is $\mathbf{v}$-recurrent. By definition of a leak witness, if we prove that (a) $\mathbf{v}_+(r',q) = 1$, (b) $\mathbf{v}(q,r') = 0$ then $(\mathbf{v},\mathbf{v}_+)$ is a leak witness.

We first prove (a). Let $\eta = p_{\min}^{3\cdot 2^{4|Q|^2}}$ and $K = |\mathcal{G}_+|!$. Then:

$$\begin{aligned}
&\mathbf{v}(r,r') = 1 && \text{(by definition of } r') \\
&\implies \mathbf{u}_N^K(r,r') = 1 && \text{(by definition of } \mathbf{v}) \\
&\implies \mathbf{u}_n^K(r,r') = 1, \text{ for infinitely many } n && \text{(by definition of } N) \\
&\implies u_n^K(r,r') \geq \eta^K, \text{ for infinitely many } n && \text{(by (27))} \\
&\implies u^K(r,r') \geq \eta^K && \text{(because } u = \lim_n u_n) \\
&\implies r' \text{ is } u\text{-recurrent} && \text{(because } r \text{ is } u\text{-recurrent}) \\
&\implies \exists l, u^l(r',r) > 0 && (30) \\
&\text{(because } r \text{ and } r' \text{ are in the same class of } u\text{-recurrence)} \\
&\implies \exists l, u^{l+1}(r',q) > 0 && \text{(because } u(r,q) > 0) \\
&\implies \exists l, \exists N', \forall n \geq N', u_n^{l+1}(r',q) > 0 && \text{(because } u = \lim_n u_n) \\
&\implies \exists N', \forall n \geq N', u_n(r',q) > 0 && \text{(the } u_n \text{ are idempotent)} \\
&\implies \exists N', \forall n \geq N', \mathbf{u}_{+,n}(r',q) = 1 && \text{(by (26))} \\
&\implies \mathbf{u}_{+,N}(r',q) = 1 && \text{(by definition of } N) \\
&\implies \mathbf{v}_+(r',q) = 1 && \text{(according to (29)).}
\end{aligned}$$

Now we prove b). By contradiction, suppose that $\mathbf{v}(q, r') = 1$. Then:

$$
\begin{aligned}
&\mathbf{v}(q, r') = 1 \\
&\implies \mathbf{u}_N^K(q, r') = 1 && \text{(by definition of } \mathbf{v}) \\
&\implies \mathbf{u}_n^K(q, r') = 1, \text{ for infinitely many } n && \text{(by definition of } N) \\
&\implies u_n^K(q, r') \geq \eta^K, \text{ for infinitely many } n && \text{(by (27))} \\
&\implies u^K(q, r') \geq \eta^K, && \text{(because } u = \lim_n u_n) \\
&\implies r' \text{ is reachable from } q \text{ in } \mathcal{M}_u \\
&\implies r \text{ is reachable from } q \text{ in } \mathcal{M}_u && \text{(according to (30))}
\end{aligned}
$$

which contradicts (25).

This completes the proof of b), thus $(\mathbf{v}, \mathbf{v}_+)$ is a leak witness, which completes the proof of Lemma 24. □

E A few leaktight automata

Proposition 12 *Deterministic automata, hierarchical probabilistic automata and ♯-acyclic automata are leaktight.*

Proof. It is obvious that deterministic automata are leaktight. We give an algebraic proof. For deterministic automata the iteration operation has no effect on limit-words. As a consequence, the extended Markov monoid only contains pair $(\mathbf{u}, \mathbf{u})$ whose both components are equal, and none of them can be a leak witness. The characterization given by Theorem 5, allows us to conclude that deterministic automata are leaktight.

The proof for hierarchical automata is given in Proposition 25.

The proof for ♯-acyclic automata is given in Proposition 26. □

Proposition 13 *The leaktight property is stable by parallel composition and synchronized product.*

Proof. For parallel product, this is easy. Let i be the new initial state. If there is a leak in $\mathcal{A}||\mathcal{B}$ from a state $q \neq i$ then this a leak either in $\mathcal{A}$ or $\mathcal{B}$. There can be no leak $(u_n)_{n\in\mathbb{N}}$ from i because i is u-recurrent only for those words u that are written with letters stabilizing i.

For synchronized product, the proof is easy as well: the extended Markov monoid of the synchronized product $\mathcal{A} \times \mathcal{B}$ is the product of the extended Markov monoids of $\mathcal{A}$ and $\mathcal{B}$. If there was a leak in $\mathcal{A} \times \mathcal{B}$, then according to Theorem 11 there would be a leak witness $(\mathbf{u}, \mathbf{u}_+) = ((\mathbf{u}_\mathcal{A}, \mathbf{u}_\mathcal{B}), (\mathbf{u}_{+,\mathcal{A}}, \mathbf{u}_{+,\mathcal{B}}))$ in the extended Markov monoid of $\mathcal{A} \times \mathcal{B}$ from a state $(r_\mathcal{A}, r_\mathcal{B})$ to a state $(q_\mathcal{A}, q_\mathcal{B})$. Then $r_\mathcal{A}$ is $\mathbf{u}_\mathcal{A}$-recurrent and $\mathbf{u}_{+,\mathcal{A}}(r_\mathcal{A}, q_\mathcal{A}) = 1$ thus since $\mathcal{A}$ is leaktight $\mathbf{u}_\mathcal{A}(q_\mathcal{A}, r_\mathcal{A}) = 1$. Similarly, $\mathbf{u}_\mathcal{B}(q_\mathcal{B}, r_\mathcal{B}) = 1$ thus $\mathbf{u}((q_\mathcal{A}, q_\mathcal{B}), (r_\mathcal{A}, r_\mathcal{B})) = 1$ hence a contradiction. □

E.1 Leaktight automata strictly contain hierarchical automata

The class of hierarchical automata has been defined in [CSV09].

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 is a mapping rank : $Q \to [1, \ldots, l]$ such that $\forall a \in A, \forall s, t \in Q$ such that $a(s, t) > 0$, $\text{rank}(t) \geq \text{rank}(s)$ and the set $\{t \mid a(s, t) > 0, \text{rank}(t) = \text{rank}(s)\}$ is either empty or a singleton.

Proposition 25. *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 automata, for every state r:

$$(r \text{ is } \mathbf{u}\text{-recurrent}) \implies (\forall q \neq r, \mathbf{u}_+(r,q) = 0) \ . \tag{31}$$

Property (31) obviously holds for base elements $(\mathbf{a}, \mathbf{a})$.

Property (31) is stable by product: let $(\mathbf{u}, \mathbf{u}_+)$ and $(\mathbf{v}, \mathbf{v}_+)$ with property (31) and let $r \in Q$ be $\mathbf{uv}$-recurrent. By definition of hierarchical automata the recurrence classes of the limit words $\mathbf{u}, \mathbf{v}$ and $\mathbf{uv}$ are singletons thus r is necessarily both $\mathbf{u}$-recurrent and $\mathbf{v}$-recurrent. According to (31), $\forall q \neq r$, $\mathbf{u}_+(r,t) = 0$ thus $\forall q \neq r, (\mathbf{u}_+\mathbf{v}_+)(r,q) = 0$.

Property (31) is obviously stable by iteration, which terminates the proof. □

The inclusion is strict, an example is given by Fig. 2.

E.2 Leaktight automata strictly contain ♯-acyclic automata

The class of ♯-acyclic automata has been defined in [GO10].

Let $\mathcal{A}$ be a probabilistic automaton, to define ♯-acyclic automata, we define an action on non-empty subsets of states. Given $S \subseteq 2^Q$ and a letter a, by definition $S \cdot \mathbf{a} = \{t \mid \exists s \in S, \mathbf{a}(s,t) = 1\}$. If $S \cdot \mathbf{a} = S$, then we define the iteration of $\mathbf{a}$: $S \cdot \mathbf{a}^\sharp = \{t \mid \exists s \in S, \mathbf{a}^\sharp(s,t) = 1\}$. Consider now the graph whose vertices are non-empty subsets of states and there is an edge from S to T if $S \cdot \mathbf{a} = T$ or $S \cdot \mathbf{a} = S$ and $S \cdot \mathbf{a}^\sharp = T$. The automaton $\mathcal{A}$ is ♯-acyclic if the unique cycles in this graph are self loops.

We extend the action on any limit-word: given $S \subseteq Q$ and a limit-word $\mathbf{u}$, by definition $S \cdot \mathbf{u} = \{t \mid \exists s \in S, \mathbf{u}(s,t) = 1\}$.

Definition 17. *$\mathcal{A}$ is not ♯-acyclic if and only if there exists S, T subsets of states, $S \neq T$, $\mathbf{u}, \mathbf{v}$ two limit-words such that $S \cdot \mathbf{u} = T$ and $T \cdot \mathbf{v} = S$.*

Proposition 14 *Deterministic automata and ♯-acyclic automata have ♯-height* 1.

Proof. For deterministic automata, this is obvious because there are no unstable idempotent in the Markov monoid so the iteration operation is useless and the ♯-heigth is actually 0.

For ♯-acyclic automata, this is a corollary of results in [GO10]: if a ♯-acyclic automaton has value 1 then there exists a sequence of letters $a_0, b_0, a_1, \ldots, a_n, b_n, a_{n+1} \in (A \cup \{\epsilon\})^*$ such that $a_0 b_0^\sharp a_1 b_1^\sharp \ldots a_n b_n^\sharp a_{n+1}$ is a value 1 witness. □

Proposition 26. *Every ♯-acyclic automata is leaktight.*

Proof. We prove that for all extended limit-word $(\mathbf{u}, \mathbf{u}_+)$, we have $(\mathbf{u}(s,t) = 0, \mathbf{u}_+(s,t) = 1) \Rightarrow$ s is transient, which implies the leaktight assumption, by induction on $\mathbf{u}$. The case $\mathbf{u} = \mathbf{a}$ is clear. Consider the case $\mathbf{u} = \mathbf{v}^\sharp$, and let s, t states such that $(\mathbf{u}(s,t) = 0, \mathbf{u}_+(s,t) = 1)$. Then either $(\mathbf{v}(s,t) = 0, \mathbf{v}_+(s,t) = 1)$ or $\mathbf{v}(s,t) = 1$ and t is transient in $\mathbf{v}$. In the first case, the induction hypothesis ensures that s is transient in $\mathbf{v}$. In the second case, s would be transient in $\mathbf{v}$. In both cases, s is transient in $\mathbf{v}$, so also in $\mathbf{u}$.

Consider now the case $\mathbf{u} = \mathbf{u}_1 \cdot \mathbf{u}_2$, and let s, t states such that $(\mathbf{u}(s,t) = 0, \mathbf{u}_+(s,t) = 1)$. Assume toward contradiction that s is recurrent in $\mathbf{u}$. Let $C = \{q \mid \mathbf{u}(s,q) = 1\}$ be the recurrence class of s, so we have $C \cdot \mathbf{u} = C$. The ♯-acyclicity implies that $C \cdot \mathbf{u}_1 = C$ and $C \cdot \mathbf{u}_2 = C$.

There are two cases: either there exists p such that $(\mathbf{u}_1(s,p) = 0, \mathbf{u}_{1_+}(s,p) = 1)$, or such that $\mathbf{u}_1(s,p) = 1$ and $(\mathbf{u}_2(s,p) = 0, \mathbf{u}_{2_+}(s,p) = 1)$. Consider the first case, and $T = C \cdot \mathbf{u}_1^\sharp$. We have $T \subsetneq C$, so $T \cdot u^\sharp = C$, which defines a ♯-cycle over C, contradiction. In the second case, let $T = C \cdot \mathbf{u}_2^\sharp$, we have $T \subsetneq C$ so $T \cdot \mathbf{u}^\sharp = C$, which defines a ♯-cycle over C, contradiction. This completes the proof. □

The inclusion is strict: Fig. 4 provides an example of leaktight automaton which is not $\sharp$-acyclic.