Université Bordeaux1

Master Thesis

Value 1 Problem for Probabilistic Automata

Author:
Edon Kelmendi[*]

Supervisor:
Hugo Gimbert[†]

June 2013

LaBRI and Université Bordeaux1
351 Cours de la libération -33405- Talence Cedex, FRANCE

[*]edon.kelmendi@gmail.com
[†]hugo.gimbert@labri.fr
http://www.labri.fr/perso/gimbert/

UNIVERSITÉ BORDEAUX1

Abstract

Value 1 Problem for Probabilistic Automata

by Edon Kelmendi

Most problems regarding probabilistic automata are undecidable, including the value 1 problem, with which we are concerned. The value 1 problem is: do there exist words with acceptance probability arbitrary close to 1? One way to cope with undecidability of this problem is to look for restricted classes of probabilistic automata for which the value 1 problem is decidable. A few such classes have been proposed in the literature. Notably *leaktight automata* in [FGO11] and *structurally simple automata* in [CT11]. It was not known whether these two classes intersect or not. Herein we compare these two classes and prove that structurally simple automata are strictly contained in leaktight automata.

After this, we report on initial research on existing positive decidability results for quantum automata and try to adapt the same techniques to probabilistic automata. We come to the conclusion that a straightforward adaptation is not sufficient to extend the class of leaktight automata.

Contents

Abstract **ii**

1 Preliminaries **1**
1.1 Introduction and previous work . . . 1
1.2 Probabilistic Automata . . . 3
1.2.1 An example . . . 4
1.2.2 Combining probabilistic automata . . . 5
1.3 Stochastic languages . . . 7
1.4 Decidable and Undecidable Problems . . . 9

2 Two Classes With Decidable Value 1 Problem **12**
2.1 Simple Automata . . . 12
2.1.1 Definitions . . . 12
2.1.2 Lemma for showing (non)simplicity . . . 16
2.1.3 Another proof of Lemma 2.6 . . . 18
2.2 Leaktight Automata . . . 19
2.2.1 Definitions . . . 19
2.2.2 Markov monoid . . . 21
2.2.3 An example . . . 22
2.2.4 Using the Markov monoid to decide value 1 . . . 24
2.2.5 Consistency . . . 26
2.2.6 Leak witness . . . 27
2.2.7 Completeness . . . 29

3 Comparison of Simple and Leaktight Automata **31**
3.1 Inclusion . . . 31
3.2 Strict inclusion . . . 38

4 Quantum automata **40**
4.1 Definitions . . . 40
4.2 Decidability results . . . 42
4.3 Application to probabilistic automata . . . 44

5 Conclusion and future work **46**

Bibliography **48**

Chapter 1

Preliminaries

1.1 Introduction and previous work

Probabilistic automata are a natural extension of deterministic finite automata, first proposed by Rabin in [Rab63]. When probabilistic automata read a letter they do not proceed to a determined next state, but instead, they go to every state of the automaton with certain probability. In the case of deterministic automata, starting at an initial state the automaton either reaches a final state when reading a word or it does not, while in a probabilistic automaton we see that it can reach a final state with a certain probability. Therefore we usually define languages accepted by a probabilistic automaton as the set of words which are accepted with a probability (strictly) bigger than a certain real number λ. We call this real number λ a *cutpoint*. If for a given probabilistic automaton there exists a fixed neighborhood of a cutpoint such that no word is accepted with probability in that neighborhood, then that cutpoint is called *isolated*. Rabin in [Rab63] shows that languages defined by probabilistic automata and isolated cutpoints are just the regular languages. But the same does not hold for non-isolated cutpoints, hence the languages defined by probabilistic automata strictly contain regular languages. Similarly to deterministic automata, probabilistic automata can be defined on finite or infinite words.

It turns out that very few problems about the languages defined by probabilistic automata are decidable, for example equivalence between two automata, while others are undecidable, this includes: equality, emptiness, existence of an isolated cutpoint, and given a cutpoint whether it is isolated or not. The last one for the special case of the cutpoint 1 is called *value 1 problem*. It basically asks the question: if there exists a sequence of finite words such that the probability to accept them gets arbitrarily close to 1.

We study this problem because probabilistic automata can be seen as partially observable Markov decision processes and stochastic games. A machine where the controller is partially blind, does not have perfect information, is very common in practice, and probabilistic automata are a good way of modeling this. And the value 1 would ask the question whether this kind of machine is controllable with arbitrarily high reliability.

Recently in [FGO11] and [CT11] two classes of probabilistic automata are defined: leaktight and structurally simple probabilistic automata respectively. For these two classes of automata the value 1 problem is decidable and they both seem to be robust classes containing other known classes with decidable value 1 problem such as #-acyclic automata defined in [GO10] and hierarchical automata in [CSV11]. In this thesis we compare these two classes of automata and show that structurally simple automata are a strict subset of the leaktight ones. Apart from reporting on existing results, this thesis gives two new contributions.

1. We give a sufficient condition for showing non-simplicity which does not depend on the *decomposition separation theorem*. For the sake of completeness we also provide the proof of the same condition that was given in [CT11].
2. The main contribution of this thesis is showing that simple automata are a proper subset of leaktight automata. We prove this by looking for a *non-simplicity witness* in the Markov monoid, and then showing that the existence of a leak implies the existence of such an element. Then using a simple example we prove that the inclusion is strict. The proof is self-contained and relies on our first contribution.

Finally we look at an interesting direction for enlarging the class of leaktight automata. The direction that we look at is *measure-once* quantum automata, for which the value 1 problem is decidable using techniques from algebraic geometry, among others. We try to adapt the same techniques in a straightforward manner to probabilistic automata, but they prove insufficient for enlarging the class of leaktight automata.

The outline of the thesis is as follows: the first chapter gives a short introduction to probabilistic automata, in the second chapter we give more context about the two classes of automata with decidable value 1 problem and in the third chapter, where the main contribution of this thesis rest, we prove that the leaktight automata contain strictly the simple ones. In the end we give a very short introduction to quantum automata and try to extend the leaktight class with the techniques used for deciding the value 1 problem for quantum automata.

1.2 Probabilistic Automata

Definition 1.1. (Probabilistic automaton) Formally a probabilistic automaton $\mathcal{A}$ is given as a tuple $\mathcal{A} = (Q, A, M_a, i, F)$ where:

- Q is a set of finite states
- A is a finite alphabet
- M_a defined for all $a \in A$ is a $|Q| \times |Q|$ stochastic matrix.
- $i \in Q$ is an initial state
- $F \subseteq Q$ is a set of final states.

A vector in $\mathbb{R}^n$, $(q_1, \ldots, q_n)$ is called stochastic if $\sum_{i=1}^{n} q_i = 1$ and $q_i \in [0, 1]$ for $i \in \{1, \ldots, n\}$. A matrix whose rows are stochastic vectors is called a stochastic matrix. Hence for some fixed $i \in 1, \ldots, |Q|$ and any $a \in A$ we have $\sum_{j=1}^{n}(M_a)_{ij} = 1$, where with $(M_a)_{ij}$ we write the element in the i-th row and j-th column of the matrix M_a. Let $Q = \{q_1, q_2, \ldots, q_n\}$. For a letter $a \in A$ the (i, j) element of M_a gives the probability to go from state q_i to state q_j with the letter a.

Remark 1.2. From the definition we see that every state $q_i \in Q$ and letter $a \in A$ there exists a state $q_j \in Q$ such that $(M_a)_{ij} > 0$. In other words no dead lock is allowed for any word in the probabilistic automaton.

Given some path, the probability that the automaton will take that path is the product of the probabilities on the edges of the path. The probability to go from state q_i to q_j is the sum of the probabilities of all the paths that go from q_i to q_j. From here, to go from state q_i to state q_j with the word aa' where $a, a' \in A$ the probability would be $\sum_{k=1}^{n}(M_a)_{ik}(M_{a'})_{kj}$. By induction we see that for a word $w = a_1 a_2 \ldots a_n$ the probability to go from state q_i to state q_j would be the usual matrix multiplication:

$$(M_{a_1} M_{a_2} \ldots M_{a_n})_{ij}.$$

Consequently for any word $w = a_1 a_2 \ldots a_n \in A^*$ we write $M_w = M_{a_1} M_{a_2} \ldots M_{a_n}$.

Let $\mathcal{D}(Q)$ be the set of all distributions on the set Q, that is all functions $d : Q \to [0, 1]$ with the property $\sum_{q \in Q} d(q) = 1$. A distribution $d \in \mathcal{D}(Q)$ can be written as a $1 \times n$ row matrix $\mathbf{d} = (d(q_1), d(q_2), \ldots, d(q_n))$. Given any distribution $\alpha \in \mathcal{D}(Q)$ written as the row vector α we see that for a word $w \in A^*$ the row vector αM_w gives the distribution induced in the automaton $\mathcal{A}$ by the word w with initial distribution α. We give this notation for such an induced distribution:

$$w(\alpha) := \alpha M_w.$$

Given some set of states $S \subseteq Q$ write with $\mathbf{S}$ as the column vector such that $(\mathbf{S})_{i1} = 1$ if and only if $q_i \in S$ otherwise it is 0. For any distribution $d \in \mathcal{D}(Q)$ and set of states $S \subseteq Q$, $\mathbf{dS}$ is the sum of distribution in the elements of S, i.e. $\sum_{q \in S} d(q)$. For the probability to reach a set of states $S \subseteq Q$ from some initial distribution $\alpha \in \mathcal{D}(Q)$ using the word $w \in A^*$ we write:

$$w(\alpha, S) := \alpha M_w \mathbf{S}.$$

If we look at the state $i \in Q$ as an initial distribution, that is a row vector that has a 1 in the position of i and 0 everywhere else we can define the probability for an automaton $\mathcal{A}$ to accept a word $w \in A^*$ as:

$$\mathcal{A}(w) := w(i, F)$$

We denote by $p_{\min}$ the smallest element of all the matrices $(M_a)_{a \in A}$.

Remark 1.3. Note that having just one state as an initial state instead of, more generally, an initial distribution, does not make any difference in the expressive power of the automaton. For instance if $\mathcal{A}$ is an automaton that has an initial distribution α we can build another automaton $\mathcal{B}$ that has one initial state i such that for all words $w \in A^*$, $\mathcal{A}(w) = \mathcal{B}(\$w)$. We add the state i and another letter to the alphabet, say \$, and from there we go with \$ to state q with probability $\alpha(q)$ and everything else the same as $\mathcal{A}$.

Now for any $\lambda \in [0, 1]$ and probabilistic automaton $\mathcal{A}$ with alphabet A we can define a language $\mathcal{L}_>(\mathcal{A}, \lambda) = \{w \in A^* \mid \mathcal{A}(w) > \lambda\}$, and similarly for other relations such as $\geq, <, \leq, =, \neq$.

Equivalently we could have given a function $\delta : Q \times A \to \mathcal{D}(Q)$, instead of the set of matrices M_a, with minor changes.

We see that deterministic finite automata are probabilistic automata where the entries in the matrices M_a are binary. Hence probabilistic automata are a very natural generalization.

1.2.1 An example

It is helpful to give labeled graphs representing probabilistic automata as in the case of other automata.

Example 1.1. *Let us give* $\mathcal{A} = (Q, A, i, \{M_a, M_b\}, F)$ *where*

- $Q = \{q_1, q_2, q_3\}$
- $A = \{a, b\}$
- $i = q_1$
- $M_a = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $M_b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix}$
- $F = \{q_3\}$

The labeled graph for this automaton would look like this:

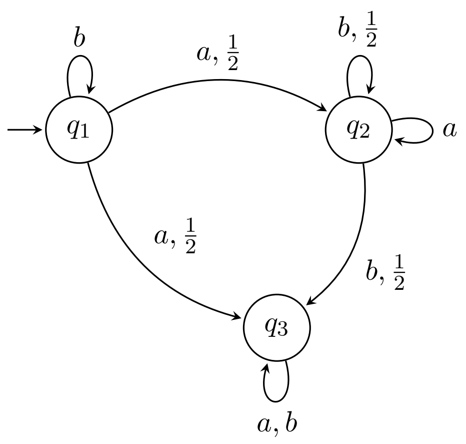

Figure 1.1: Probabilistic automaton $\mathcal{A}$

When the transition probability is 1 we omit writing it. In this example we have for the word $w = abb$*:*

$$\mathcal{A}(abb) = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{7}{8}$$

Where each term is one of the paths from q_1 *to* q_3 *labeled abb.*

1.2.2 Combining probabilistic automata

It is helpful to construct automata out of some given automata such that a particular equation holds between their functions. For example we can complement an automaton:

Theorem 1.4. *Given an automaton* $\mathcal{A}$ *we can build and automaton* $\mathcal{B}$ *such that for all words* $w \in A^*$ *we have:*

$$\mathcal{B}(w) = 1 - \mathcal{A}(w).$$

Proof. Let $\mathcal{A} = (Q, A, i, (M_a)_{a\in A}, F)$. We give $\mathcal{B}(Q, A, i, (M_a)_{a\in A}, Q \setminus F)$. We are just inverting the final states. This comes from the simple fact that if we can reach the set of final states with some probability p then we can reach the set of non-final states with probability $1 - p$. $\square$

We can also take the product of two automata:

Theorem 1.5. *Given automata $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same alphabet A we can build the automaton $\mathcal{B}$ such that for all words $w \in A^*$ we have:*

$$\mathcal{B}(w) = \mathcal{A}_1(w)\mathcal{A}_2(w).$$

Proof. We basically take the cartesian product of these two automata. Let $\mathcal{A}_1 = (Q_1, A, i_1, (M_{1a})_{a\in A}, F_1)$ and $\mathcal{A}_2 = (Q_2, A, i_2, (M_{2a})_{a\in A}, F_2)$.

We construct $\mathcal{B} = (Q, A, i, (M_a)_{a\in A}, F)$ such that $Q = Q_1 \times Q_2$, $i = (i_1, i_2)$, $(M_a)_{(q_1,q_2),(q'_1,q'_2)} = (M_{1a})_{q_1,q'_1}(M_{2a})_{q_2,q'_2}$. One can easily show that this is a stochastic matrix. And finally $F = \{(f_1, f_2) \mid f_1 \in F_1 \text{ and } f_2 \in F_2\}$. $\square$

Sometimes it might be helpful to scale down the probabilities of an automaton:

Theorem 1.6. *Given automaton $\mathcal{A}$ and a real $c \in [0, 1]$ we can build an automaton $\mathcal{B}$ such that for all words $w \in A^+$ we have:*

$$\mathcal{B}(w) = c\mathcal{A}(w).$$

Proof. Let $\mathcal{A} = (Q, A, i, (M_a)_{a\in A}, F)$. We give $\mathcal{B} = (Q \cup \{\perp\}, A, i, (M'_a)_{a\in A}, F)$. Where $(M'_a)_{i,q} = c(M_a)_{i,q}$ and $(M'_a)_{i,\perp} = 1 - c$, and all other elements of the matrices are the same. $\square$

Finally we can also sum two matrices:

Theorem 1.7. *Given automata $\mathcal{A}_1$, $\mathcal{A}_2$ over the same alphabet A and two positive reals c_1, c_2 such that $c_1 + c_2 = 1$ then we can build an automaton $\mathcal{B}$ such that for all words $w \in A^*$ we have:*

$$\mathcal{B}(w) = c_1\mathcal{A}_1(w) + c_2\mathcal{A}_2(w)$$

Proof. Let $\mathcal{A}'_1 = (Q_1, A, i_1, (M_{1a})_{a\in A}, F_1)$ and $\mathcal{A}'_2 = (Q_2, A, i_2, (M_{2a})_{a\in A}, F_2)$. We build $\mathcal{B} = (Q_1 \cup Q_2 \cup \{i\}, A, i, (M_a)_{a\in A}, F_1 \cup F_2)$. And $(M_a)_{i,q} = c_1(M_{1a})_{i_1,q} + c_2(M_{2a})_{i_2,q}$ for $q \in Q$, while for $q, q' \in Q_1$ we have $(M_a)_{q,q'} = (M_{1a})_{q,q'}$, and symmetrically for $q, q' \in Q_2$ we have $(M_a)_{q,q'} = (M_{2a})_{q,q'}$. And there are no transitions between states in Q_1 and states in Q_2, that is for $q \in Q_1$ and $q' \in Q_2$ we have $(M_a)_{q,q'} = 0$ and vice versa. $\square$

1.3 Stochastic languages

In this section we sketch two important results due to Rabin [Rab63]. After defining probabilistic automata the first natural question is: are probabilistic automata more expressive than deterministic ones? It is obvious that they are at least as expressive as deterministic ones since the latter are a special case of the former (namely the case when the matrices are binary).

An important role into answering this question is played by the notion of an isolated cutpoint.

Definition 1.8. (Isolated cutpoint) Given some automaton $\mathcal{A}$ and cutpoint $\lambda \in [0,1]$ we say that λ is isolated if there exists $\epsilon > 0$ such that for all words $w \in A^*$ we have:

$$|\mathcal{A}(w) - \lambda| > \epsilon.$$

We call some cutpoint isolated, if we can give some ϵ neighborhood of it such that no word is accepted by $\mathcal{A}$ with probability in that neighborhood.

Definition 1.9. (Stochastic language) Given some automaton $\mathcal{A}$, cutpoint $\lambda \in [0,1]$ and relation $\diamond \in \{<, \leq, >, \geq, =, \neq\}$ we write

$$\mathcal{L}_\diamond(\mathcal{A}, \lambda) = \{w \in A^* \mid \mathcal{A}(w) \diamond \lambda\}$$

As it turns out if we look at just the languages defined for isolated cutpoints, probabilistic automata do not add any expressive power to the deterministic automata.

Theorem 1.10. *([Rab63]) Let $\mathcal{A}$ be a probabilistic automaton and $\lambda \in [0,1]$ an isolated cutpoint in it. Then the language $\mathcal{L}_>(\mathcal{A}, \lambda)$ is regular.*

Proof. Since λ is an isolated cutpoint there exists $\epsilon > 0$ such that for all words $w \in A^*$ we have $|\mathcal{A}(w) - \lambda| > \epsilon$.

For two words $x, y \in A^*$ write $x \equiv y$ if and only if for all $z \in A^*$ it is true that:

$$xz \in \mathcal{L}_>(\mathcal{A}, \lambda) \iff yz \in \mathcal{L}_>(\mathcal{A}, \lambda).$$

This partitions the set of all finite words into equivalence classes. The Myhill-Nerode theorem [Ner58] says that this language is regular if and only if the number of such equivalence classes is finite.

Suppose that $x \not\equiv y$ then there exists a $z \in A^*$ such that $xz \in \mathcal{L}_>(\mathcal{A}, \lambda)$ and $yz \notin \mathcal{L}_>(\mathcal{A}, \lambda)$. Consequently this means that $\mathcal{A}(xz) > \lambda + \epsilon$ and $\mathcal{A}(yz) \leq \lambda - \epsilon$. Adding these two inequalities we get

$$\mathcal{A}(xz) - \mathcal{A}(yz) > 2\epsilon.$$

Unwrapping the definition of such a function we get:

$$(\mathbf{i}M_x - \mathbf{i}M_y)M_z\mathbf{F} > 2\epsilon$$

Since $M_z\mathbf{F}$ is a column vector where the elements are at most 1, we have:

$$\|\mathbf{i}M_x - \mathbf{i}M_y\|_1 \geq (\mathbf{i}M_x - \mathbf{i}M_y)M_z\mathbf{F} > 2\epsilon$$

x and y were arbitrary, therefore we can not have an infinite number of them because the 1-norm of their distributions is bounded from below. □

But by removing this restriction for isolated cutpoints in [Rab63] they show that the number of stochastic languages is uncountable (while regular languages are not). This can be seen by giving these two matrices:

$$M_0 = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, M_1 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}$$

Construct an automaton $\mathcal{A}$ with states q_1, q_2 where q_1 is the initial state and q_2 is the final state, and alphabet $A = \{0, 1\}$ with the corresponding matrices given above. It can be verified that given a word $w = a_1a_2 \ldots a_n$ we have $\mathcal{A}(w) = 0.a_na_{n-1} \ldots a_1$ in binary. This is called the *2-adic* language. In particular for an irrational cutpoint this language is not regular. This in turn also shows that the set of stochastic languages is uncountable.

Even if we restrict ourselves to rational cutpoints, the languages defined, are not necessarily regular, as it can be seen from the following example. Let $A = \{a, b, c\}$ be the alphabet. The language over A, $L = \{cw \in (A \setminus \{c\})^* \mid w \text{ has more } a\text{'s than } b\text{'s}\}$[1] is not regular. We show that this is accepted by a PA with cutpoint $\frac{1}{2}$.

[1]That is $N_a(w) \geq N_b(w)$ where $N_a(w)$ is the number of a's in w.

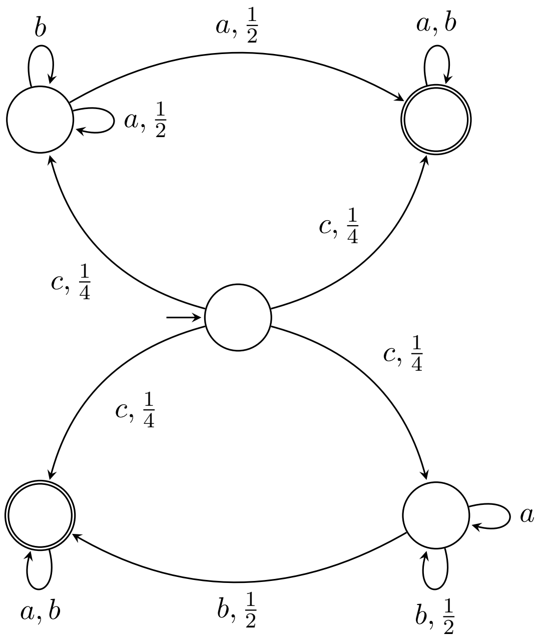

FIGURE 1.2: The language of this automaton at cutpoint $\frac{1}{2}$ is not regular.

Denote by $\mathcal{A}$ the PA in Figure 1.2. Then $\mathcal{L}_{\geq}(\mathcal{A}, \frac{1}{2}) = L$, because of the following. After playing c in the initial state we have $\frac{1}{2}$ probability in final states, playing a's increases this amount, and playing b's decreases it. The first time you play an a you increase it by $\frac{1}{8}$, and the first time you play a b, you decrease it by the same amount.

1.4 Decidable and Undecidable Problems

In this section we give a short survey on a few decidability results for stochastic languages.

Problem 1.1. *(Emptiness problem) Given an automaton $\mathcal{A}$ and a cutpoint λ decide whether $\mathcal{L}_{\diamond}(\mathcal{A}, \lambda) = \emptyset$. Where $\diamond \in \{<, \leq, >, \geq, =, \neq\}$.*

Note that $\mathcal{L}_{<}(\mathcal{A}, \lambda) = \emptyset \iff \mathcal{L}_{\geq}(\mathcal{A}, \lambda) = A^*$, and similarly for other relations. Also from Theorem 1.4 we can reduce the emptiness problem for the language $\mathcal{L}_{<}(\mathcal{A}, \lambda)$ to $\mathcal{L}_{\geq}(1 - \mathcal{A}, \lambda)$, and vice versa.

Theorem 1.11. *([Paz71, GO10]) Problems in Problem 1.1 are undecidable.*

Proof. (Sketch) We sketch the proof given in [GO10]. First we need an undecidable problem:

Problem 1.2. *(PCP) Let $\phi_1, \phi_2 : A \to \{0,1\}^*$ be two morphisms ($\phi_i(w_1w_2) = \phi_i(w_1)\phi_i(w_2)$). Decide whether there exists $w \in A^+$ such that $\phi_1(w) = \phi_2(w)$.*

From an instance of Problem 1.2 we can construct a probabilistic automaton $\mathcal{A}$ such that there exists a word $w \in A^*$ that $\mathcal{A}(w) = \frac{1}{2}$ if and only if $\phi_1(w) = \phi_2(w)$. The idea is to see the output of ϕ_1, ϕ_2 as elements in $[0,1]$.

After this we can reduce the emptiness of $\mathcal{L}_=(\mathcal{A}, \lambda)$ to emptiness of $\mathcal{L}_\geq(\mathcal{A}', \lambda')$. This is easily done from Theorems 1.4 and 1.5, we build an automaton $\mathcal{B}$ such that $\mathcal{B}(w) = \mathcal{A}(w)(1 - \mathcal{A}(w))$, and we also have that for a real number x, $x = \frac{1}{2} \iff x(1-x) \geq \frac{1}{4}$. By adding a few states to $\mathcal{B}$ we can construct another automaton such that the emptiness problem of $\mathcal{L}_\geq$ can be reduced to the emptiness problem of $\mathcal{L}_>$. □

Another undecidable problem, and the one we are interested in is:

Problem 1.3. *Given an automaton $\mathcal{A}$ and a cutpoint λ. Decide whether λ is an isolated cutpoint.*

The special case of Problem 1.3 where $\lambda = 1$ is called *the value 1 problem.*

Theorem 1.12. *([BMT77, GO10]) Problem 1.3 is undecidable.*

Proof. (Sketch) In [BMT77] they showed undecidability for any cutpoint $0 < \lambda < 1$ and in [GO10] they show it also for $\lambda = 0$ and $\lambda = 1$. We provide a sketch of the proof in [GO10]. The idea is to reduce the emptiness problem for $\mathcal{L}_>(\mathcal{A}', \lambda')$ to the existence of an isolated cutpoint. This can be done by the following automaton.

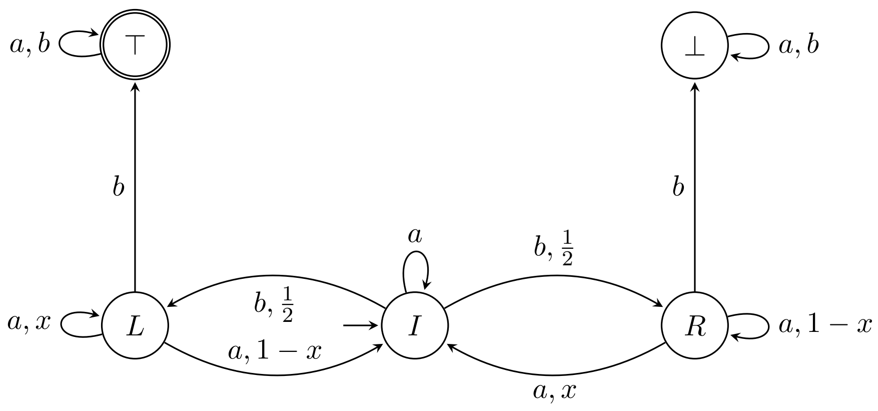

FIGURE 1.3: Automaton that has a value 1 if and only if $x > \frac{1}{2}$

$\top$ is the the only state in F. It can be shown that this automaton has value 1 if and only if $x > \frac{1}{2}$, in which case there exists a sequence of words whose probability to reach $\top$ gets arbitrarily close to 1.

Then the idea is to put, instead of the node L, an arbitrary automaton $\mathcal{A}$ therefore reducing the value 1 problem of this constructed automaton to the emptiness problem of $\mathcal{L}_{>}(\mathcal{A}, \frac{1}{2})$. $\square$

In the remainder of the thesis we look upon two classes of automata for which the value 1 problem is decidable. But before we go on to that we show that there are some problems which are decidable for probabilistic automata most notably the equivalence between two automata:

Problem 1.4. *Given two automata $\mathcal{A}$ and $\mathcal{B}$, decide whether for all $w \in A^*$ we have $\mathcal{A}(w) = \mathcal{B}(w)$.*

Theorem 1.13. *([SS61, Tze92]) The problem 1.4 is decidable.*

We conclude this chapter with an open problem:

Problem 1.5. *Given an automaton $\mathcal{A}$ and an isolated cutpoint λ. Decide whether $\mathcal{L}_{>}(\mathcal{A}, \lambda) = \emptyset$.*

Note that we are given an isolated cutpoint, but we do not know the ϵ neighborhood in which the cutpoint λ is isolated, otherwise the problem would become decidable.

Chapter 2

Two Classes With Decidable Value 1 Problem

2.1 Simple Automata

In [CT11] they consider probabilistic automata on infinite words, and among other things they construct a subclass of probabilistic automata called *structurally simple automata* for which the value 1 problem is decidable. They do this by relying on a result on non-homogeneous Markov chains called the *decomposition separation theorem*.

We proceed by giving the definition of a simple automaton, and a Lemma which gives a sufficient condition for non-simplicity, from [CT11]. A contribution that we give is another proof for this condition, in Lemma 2.8 which does not depend on the decomposition separation theorem. The rest of the thesis depends only on Lemma 2.8.

2.1.1 Definitions

First, since we are dealing with infinite words we find it useful to introduce the following notation. Let $w = a_1 a_2 \cdots \in A^\omega$ an infinite word, we write $w[i,j]$, $i \leq j$ for the sub-word $a_i, a_{i+1}, \ldots, a_j$. Also as a shorthand for $w[1,n]$ we write $w_{\leq n}$. Given some finite word $w \in A^*$ we write $|w|$ for the length of w.

We proceed with some definitions before introducing the decomposition separation theorem.

Definition 2.1. (Markov chain) A Markov chain is a sequence of random variables $X_1, X_2, \ldots$ taking values in a finite set Q with the Markov property:

$$\mathbb{P}(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = \mathbb{P}(X_{n+1} = x \mid X_n = x_n).$$

Given $n \in \mathbb{N}$ define the $|Q| \times |Q|$ matrix M_n called the transition matrix in the following way: $M_n(q, q') = \mathbb{P}(X_{n+1} = q' \mid X_n = q)$. The Markov chain is called *homogeneous* if M_n does not depend on n otherwise it is called *non-homogeneous.*

The Markov property says that the next state depends only on the state that we are in now, not the ones before it.

The infinite word $w \in A^\omega$ induces a non-homogeneous Markov chain on some probabilistic automaton $\mathcal{A}$. Sometimes this Markov chain is called a process. It is non-homogeneous because at different times n the transition matrices depend on which letter we are reading at that time n, and changes accordingly.

Given an automaton $\mathcal{A}$ and initial distribution α the word $w \in A^\omega$ also induces a probability distribution on the set of all infinite runs Q^ω. Call an element of Q^ω *an infinite run.* Therefore an infinite run is a run $q_1, q_2, \ldots$ where $q_1 \in Q, q_2 \in Q, \ldots$. A *positive run* is an infinite run that has strictly positive probability to happen on $\mathcal{A}$ with the word w and initial distribution α.

Definition 2.2. (Jets) A jet is a sequence $(J_n)_{n \in \mathbb{N}}$ where $J_k \subseteq Q$ for each $k \in \mathbb{N}$. A decomposition of an infinite run into jets, is a tuple of jets $(J^0, J^1, \ldots, J^c)$ for some $c \geq 1$ such that for all $k \in \mathbb{N}$, $(J^0_k, J^1_k, \ldots, J^c_k)$ is a partition of Q.

Decomposing an infinite run, or a set of infinite runs into c jets means that we give a partition of Q into c subsets, at every step, where by step we mean reading a letter of the word. Note that these subsets may be empty.

Definition 2.3. (Mixings) Given an automaton $\mathcal{A}$ a jet J and a word $w \in A^\omega$ we say that J is a *mixing* for the word w if for all $m \in \mathbb{N}$, $q, q' \in Q$ and sequences $(q_i)_{i \in \mathbb{N}}$ that belong to J (meaning for all $n \in \mathbb{N}$, $q_n \in J_n$).

If

$$\lim_{n \to \infty} \mathbb{P}[X_n = q_n \mid X_m = q] > 0 \text{ and } \lim_{n \to \infty} \mathbb{P}[X_n = q_n \mid X_m = q'] > 0,$$

then

$$\lim_{n \to \infty} \frac{\mathbb{P}[X_n = q_n \mid X_n \in J_n \ \wedge \ X_m = q]}{\mathbb{P}[X_n = q_n \mid X_n \in J_n \ \wedge \ X_m = q']} = 1,$$

where with X_n we write the n-th random variable of the induced Markov chain.

A jet is mixing with a word if for any given sequence of states that belongs to the jet the probability to stay on this sequence does not depend on what state we were at some step. Whether at step m we are at the state q or q' in the long run does not effect the probability that the infinite run will be the given sequence.

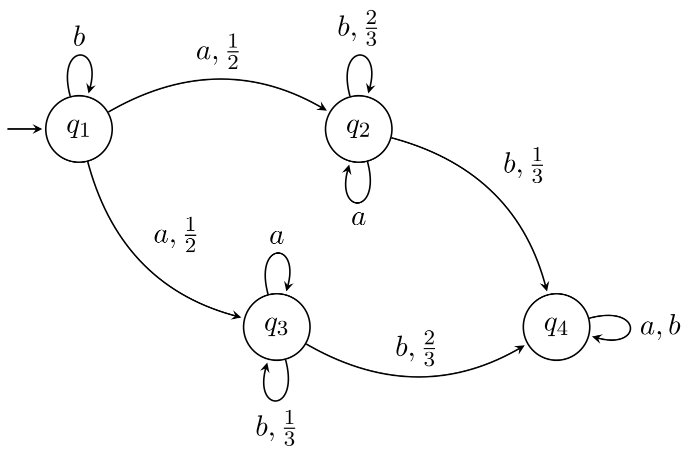

FIGURE 2.1: Mixing example

In Figure 2.1 we see that the jet $J_n = q_4$ for all $n \in \mathbb{N}$ is not a mixing for the word aba^ω, reason being if we fix $m = 2$ and take $q = q_2$ and $q' = q_2$, we see that it does depend in which one of the two states q_2 or q_3 we are at the step $m = 2$, because they go with different probabilities to q_4 hence the ratio in the definition above will not converge to 1, but to either $\frac{1}{2}$ or 2.

Theorem 2.4. *(Decomposition separation theorem, [BF64, Coh89, Son96]) Let $\mathcal{A}$ be a probabilistic automaton and $w \in A^\omega$, let $W \subseteq Q^\omega$ be the set of positive runs in $\mathcal{A}$ with w and initial distribution α. Then there exists a decomposition of W into jets $(J^0, J^1, \ldots, J^c)$ for some $c \geq 1$, $N \in \mathbb{N}$ and $k \in \{1, 2, \ldots, c\}$ such that for all $n \geq N$*

$$q_1 q_2 \ldots q_N \ldots \in W \implies q_N \in J^k_N \ \wedge \ q_n \in J^k_n.$$

Also the jets $J^1, J^2, \ldots, J^c$ are mixing for the word w.

We have a decomposition (into jets that are mixing) of positive infinite runs for the word w and initial distribution α such that for every infinite run the following holds: it eventually ends up in one of the jets $J^1, \ldots J^c$. Note that it can not end up in J^0. This means that we can not "move" between jets infinitely often, but instead the run will go into one of the jets written above and never go out of it. Entering one jet and never leaving it happens with probability one, for all the possible infinite runs, they must enter one of the jets and never leave it.

In Example 1.1 for the word b^ω we see that we can take $J_n^0 = \{q_2, q_3\}$ and $J_n^1 = \{q_1\}$ for all $n \in \mathbb{N}$. Since there is only one run and it stays in q_1. While for words which contain an a somewhere, we can take $J_n^0 = \{q_1\}$ and $J_n^1 = \{q_2, q_3\}$ for all $n \in \mathbb{N}$. Again the run after reading the a will go to one of the states in J^1 and never leave this jet. Both of these jets are mixing for trivial reasons.

We have given in these examples jets that are constant sequences, this is not a necessary condition, we have done so for simplicity.

Definition 2.5. (Simple automaton, [CT11]) Let $\mathcal{A}$ be a probabilistic automaton with state space Q, α an initial distribution and $w \in A^\omega$ an infinite word. The process induced on Q by w is called simple if and only 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, B_k) is a partition of Q,
2. for all $k \in \mathbb{N}$ and all $q \in A_k$ we have $w_{\leq k}(\alpha, q) > \lambda$,
3. $\lim_{n\to\infty} w_{\leq n}(\alpha, B_n) = 0$.

If for every word $w \in A^\omega$ the induced process in $\mathcal{A}$ is simple we call it a simple automaton.

Intuitively an infinite word w induces a simple process in the automaton $\mathcal{A}$ if and only if there exists some lower bound λ such that the states on which the distribution is below this lower bound (after a finite number of steps), converge to 0.

Hence we can at each step partition the set of states into two sets A_n and B_n such that the distribution on the latter shrinks in the next steps.

The process induced by the word w is not simple if there exists at least one state $q \in Q$ such that $w_n(\alpha, q)$ goes arbitrarily close to 0 and then resets to some constant, and this is repeated infinitely many times. The problem here being that q will be an element of B_n infinitely many times since it always goes gradually to 0[1], but at all these times it has some positive probability, hence (3) in Definition 2.5 cannot be fulfilled.

To illustrate better the concept of a simple automaton we provide a couple of examples.

Example 2.1. *In the following figure we present a simple automaton, this means that every word in the automaton induces a simple process.*

[1]But it does not reach 0.

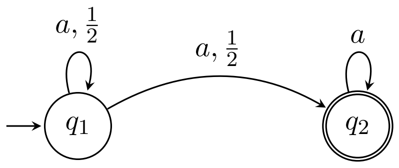

FIGURE 2.2: A simple automaton

Because the alphabet has only one letter, this is actually a homogeneous Markov chain, and the only infinite word we can have here is a^ω. We see that $a^n(q_1, q_1)$ can not be bounded from below, hence the state q_1 will eventually always be an element of the jet B_n. And the limit of $a^n(q_1, q_1)$ converges to 0 therefore (3) in Definition 2.5 holds.

Example 2.2. *In the following figure we present a nonsimple automaton, this means that there exists a word that induces a nonsimple process.*

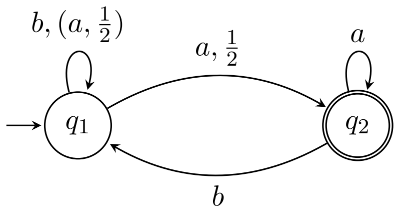

FIGURE 2.3: A nonsimple automaton

Let $w_n = a^n b$. We see that $a^n(q_1, q_1) = \frac{1}{2^n}$, hence by reading a's we can make the distribution on q_1 to go arbitrarily close to 0. But $w_n(q_1, q_1) = 1$, this is the "resetting" that we mentioned earlier. Hence for the infinite word $\rho = w_1 w_2 \dots$ we see that we can get the distribution on q_1 to go arbitrarily close to 0 and then reset to 1 and do this infinitely often. If the process induced by the word ρ was simple we would have a bound λ such that the conditions in Definition 2.5 would hold. Since by reading a's we can get the distribution on q_1 to go arbitrarily close to 0, the state q_1 should be an element of the jet B_n infinitely often. But the distribution on q_1 does not converge to 0 because it resets and repeats. Hence (3) in Definition 2.5 can not be fulfilled.

2.1.2 Lemma for showing (non)simplicity

We make the techniques that we used on the previous example more precise in the following lemmas.

Lemma 2.6. *([CT11]) Let $\mathcal{A}$ be a probabilistic automaton and $w \in A^\omega$ an infinite word that induces a simple process in $\mathcal{A}$ with bound λ and α an initial distribution, then there exists $N \in \mathbb{N}$, $c \geq 1$ and a decomposition of the set of the positive infinite runs into jets $(J^0, \dots, J^c)$ such that:*

1. *for all $i \in \{1, \ldots, c\}$, $k \in \mathbb{N}$ and all $q \in J_k^i$ we have $w_{\leq k}(\alpha, q) > \lambda$*

2. $\lim_{n\to\infty} w_{\leq n}(\alpha, J_n^0) = 0$

3. *for all $n_2 > n_1 \geq N$ and $i \in \{1, \ldots, c\}$ we have Supp$(w[n_1 + 1, n_2](J_{n_1}^i)) \subseteq J_{n_2}^i$*

If the process is simple we see that the sets $(A_n)_{n\in\mathbb{N}}$ can be decomposed into *non-communicating* jets. Meaning that once a run has entered one of them it never leaves it. Notice that the first part of the lemma is not true in general, only in the case of simple automata. The distribution of the whole jet can be bounded according to Theorem 2.4, but not the distribution of the states.

Proof. By Theorem 2.4 there exists $\lambda > 0$ and a decomposition of the set of infinite runs into jets $(K^0, K^1, \ldots, K^d)$ for some $d \geq 1$. For all $i \in \{1, \ldots, d\}$ and $n \in \mathbb{N}$ define $J_n^i = \{q \in K_n^i \mid w_{\leq n}(\alpha, q) > \lambda\}$ and $J_n^0 = \bigcup_{j=1}^d (K_n^j \setminus J_n^j) \cup K_n^0$. We claim that the decomposition $(J^0, J^1, \ldots, J^d)$ fulfills the three properties written on the statement of the Lemma. First we see that for all $n \in \mathbb{N}$ the tuple of jets $(J_n^0, J_n^1, \ldots, J_n^d)$ is indeed a partition of Q.

For (1) it is immediate from the definition of $J^1, J^2, \ldots, J^d$ and we get (2) from the fact that w induces a simple word there must be jets $(A_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$, by definition $J_n^0 \setminus K_n^0 \subseteq B_n$ for all $n \in \mathbb{N}$ because they are below λ. From Theorem 2.4 we know that no run ends up in K^0 with strictly positive probability. Hence eventually the distribution to go to this jet goes to 0.

In order to show (3), by contradiction suppose that for all $N \in \mathbb{N}$ there exist $n_2 > n_1 \geq N$ and $i \in \{1, \ldots, d\}$ such that $\text{Supp}(w[n_1 + 1, n_2](J_{n_1}^i)) \not\subseteq J_{n_2}^i$. Because we can take N arbitrarily large and Q is finite there must exist $q \in J_n^i$ and $q' \in J_{n+1}^j$ for infinitely many $n \in \mathbb{N}$ with $i \neq j$ such that $a_n(q, q') > 0$ where a_n is the n-th letter of w. This contradicts Theorem 2.4 because when the run enters J^i it's not staying in this jet with probability 1, because for an infinite amount of proceeding steps there is some probability for leaving J^i and going to J^j. □

Lemma 2.6 describes what happens when we look at the decomposition of the set of infinite runs into jets if the process is simple.

A corollary of Lemma 2.6 provides an easy way to show the simplicity of a process.

Corollary 2.7. *Let $w \in A^\omega$ be a word which induces a simple process in $\mathcal{A}$ and α an initial distribution. If there exist q, $q' \in Q$ and $\gamma > 0$ such that for infinitely many*

$n \in \mathbb{N}$, $w_{\leq n}(\alpha, q) > \gamma$ and $w[n+1, n+k_n](q, q') > 0$ for some $k_n \in \mathbb{N}$ then there exists $\gamma' > 0$ such that $w_{\leq n+k_n}(\alpha, q') > \gamma'$.

Proof. Since w induces a simple process we use Lemma 2.6 and let $J = (J^0, J^1, \ldots, J^c)$ be the decomposition. Since for an infinite number of $n \in \mathbb{N}$, $w_{\leq n}(\alpha, q) > \gamma$ then by definition of J it must appear infinitely many times in one of the jets $J^1, \ldots, J^c$. And using (3) of Lemma 2.6 by instantiating $n_1 = n$ and $n_2 = n + k_n$ we get that the run should stay in the same jet and every state of this jet should be bounded hence $w_{\leq n+k_n}(\alpha, q') > \gamma'$. □

Intuitively if we can give a lower bound on the distribution for some state q on infinitely many steps, and we also observe that we can go to some other state q' from these steps then there exists a lower bound on the distribution of the state q' also, if the process is simple.

By taking the contraposition of the Corollary above we can show for a particular word, that it does not induce a simple process and from here that the automaton itself is not simple. We can apply it to Example 2.2 and the word $\rho = w_1 w_2 \cdots \in A^\omega$ where $w_n = a^n b$. We have: $\rho_n = w_1 w_2 \ldots w_n(q_1, q_1) \geq 1$ because whenever we read a b we go back to q_1 with probability 1. But then we have $w[|\rho_n| + 1, |\rho_n| + n + 1] = a^{n+1}$ and $a^{n+1}(q_1, q_1) < \frac{1}{2^{n+1}}$, and there exists no lower bound on it.

2.1.3 Another proof of Lemma 2.6

We can give another proof of the contraposition of the Corollary 2.7 that does not rely on Theorem 2.4. Since this is easier to apply for showing the non-simplicity of processes and to demonstrate that we do not need to rely on Theorem 2.4, we give the proof in the following.

Lemma 2.8. *Let $w \in A^\omega$ and α an initial distribution. Suppose there exist $q, q' \in Q$ and $\gamma > 0$ such that for infinitely many $n \in \mathbb{N}$, $w_{\leq n}(\alpha, q) > \gamma$ and $w[n+1, n+k_n](q, q') > 0$ for some $k_n \geq 1$. If $\liminf_n w_{n+k_n}(\alpha, q') = 0$ then the process induced by w is not simple.*

Proof. Assume that w induces a simple process with bound λ. From the hypothesis and definition of a simple process, for infinitely many $n \in \mathbb{N}$, $q \in A_n$ and $q' \in B_{n+k_n}$. Since $w[n+1, n+k_n](q, q') > 0$ there exist states $t, t' \in Q$ and a sequence of integers $(j_n)_{n \in \mathbb{N}}$ such that for infinitely many $n \in \mathbb{N}$ we have $t \in A_{n+j_n}$ and $t' \in B_{n+j_n+1}$, and there exists a transition between these two states labeled w_{n+j_n}[2]. This is a contradiction because

[2]There is always going to be some state in the A sets connected to some state in the B set, but since $|Q|$ is finite, there are always two such sets, for which this repeats infinitely often.

when $t' \in B_{n+j_n+1}$ we have $w_{\leq n+j_n+1}(\alpha, t') > \lambda p_{\min}$ but by definition of a simple process $\lim_{n\to\infty} w_{\leq n+j_n+1}(\alpha, B_{n+j_n+1}) = 0$. □

In [CT11] the authors go on defining another class of automata which are called *structurally simple* automata, and which form a subset of simple automata. The value 1 is decidable for this subclass of automata. In the following we show that the value 1 problem is decidable for the whole class of simple automata (not just structurally simple) because they are a strict subset of leaktight automata.

2.2 Leaktight Automata

We now present the class of automata known as *leaktight* given in [FGO11].

2.2.1 Definitions

Given a probabilistic automaton $\mathcal{A}$, a finite word $w = a_1 a_2 \ldots a_n \in A^*$ induces a homogeneous Markov chain (see Definition 2.1) with transition matrix $M_w = M_{a_1} M_{a_2} \ldots M_{a_n}$. We say that a state $q' \in Q$ is reachable from a state $q \in Q$ in M_w if there exists $k \in \mathbb{N}$ such that $(M_w)^k_{q,q'} > 0$.

Definition 2.9. (Recurrent states) Given a homogeneous Markov chain with transition matrix M we say that the state $q \in Q$ is recurrent in M if the following holds: for all states $q' \in Q$ if q' is reachable from q in M then q is reachable from q' in M.

Given an automaton $\mathcal{A}$ and a word $w \in A^*$ then for a state $q \in Q$ we say that it is w-recurrent if it is recurrent in M_w[3]

A state that is not recurrent is called *transient*.

When a sequence of words in the limit case turns a subset of states into a recurrence class (a set of mutually reachable recurrent states) but for every element of the sequence some probability leaves this recurrence class, we say that we have a leak. Before giving the formal definition let us first define idempotent words and give a lemma in order to motivate the definition better.

Definition 2.10. (Idempotent words) A word $w \in A^*$ is called idempotent if for all states $q, q' \in Q$

$$w(q,q') > 0 \iff w^2(q,q') = ww(q,q') > 0$$

[3]The homogeneous Markov chain induced by the word w that has the transition matrix M_w.

This means that if we look at the positive reachability graph [4] for this word it is identical to the positive reachability graph of the words w^k for $k \in \mathbb{N}$.

Lemma 2.11 ([FGO11])**.** *Let w be an idempotent word and $q \in Q$ a state. Then q is w-recurrent if for all states $q' \in Q$*

$$w(q, q') > 0 \implies w(q', q) > 0$$

Proof. Immediately follows from the fact that w is an idempotent word, so for all states $s, s' \in Q$ if s' is reachable from s in M_w it is reachable in one step: $(M_w)_{s,s'} > 0$. □

Definition 2.12 (Leak)**.** A leak from a state $r \in Q$ to a state $t \in Q$ is a sequence of idempotent words $(u_n)_{n \in \mathbb{N}}$ such that:

1. for every state $q, q' \in Q$ the sequence $(u_n(q, q'))_{n \in \mathbb{N}}$ converges to some value $u(q, q')$. Denote by M_u the Markov chain with values $(u(q, q'))_{q,q' \in Q}$,
2. the state r is recurrent in M_u,
3. for all $n \in \mathbb{N}$, $u_n(r, t) > 0$,
4. r is not reachable from t in M_u.

An automaton that has no leaks is called *leaktight*.

Example 2.3. *Consider the automaton given in the following figure:*

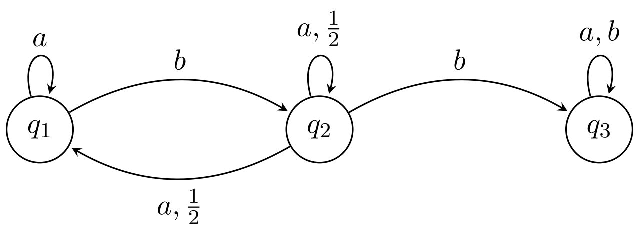

Figure 2.4: An automaton with a leak

and the sequence of words: $(a^n b)_{n \in \mathbb{N}}$. Let $u(q, q') = \lim_{n \to \infty} a^n b(q, q')$ for all $q, q' \in Q$. It can be shown that the graph in the following figure corresponds to u:

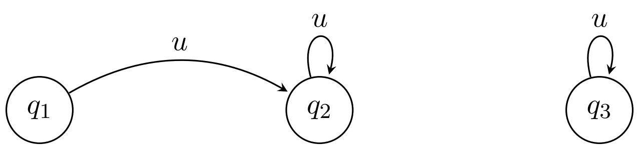

Figure 2.5: The limit case for the sequence of words $(a^n b)_{n \in \mathbb{N}}$.

[4]The graph where the vertices are the states and there is an edge from q to q' if there is some strictly positive probability to go from q to q'.

Firstly we see that the u is idempotent. Now obviously q_2 is u-recurrent, and it is not reachable by q_3. But we have that for all $n \in \mathbb{N}$ $a^n b(q_2, q_3) > 0$. Hence the sequence of words $(a^n b)_{n \in \mathbb{N}}$ is a leak from the state q_2 to the state q_3 and that the automaton given in Figure 2.4 is not leaktight.

2.2.2 Markov monoid

The Markov monoid is a means of abstracting the behavior of sequences of words in the automaton where we only care about the reachability with strictly positive probability.

The elements of the Markov monoid are called limit words:

Definition 2.13 (Limit word)**.** A limit word[5] $\mathbf{u}$ is a map $\mathbf{u} : Q \times Q \to \{0, 1\}$, where with Q we write the set of states of the automaton. Hence it also can be seen as a directed graph where the nodes are the set of states and there is a transition from q to q' if and only if $\mathbf{u}(q, q') = 1$.

The binary operator in this monoid is concatenation:

Definition 2.14 (Concatenation of limit words)**.** Let $\mathbf{u}_1$, $\mathbf{u}_2$ be two limit words, we define the concatenation[6] of them $\mathbf{u} = \mathbf{u}_1\mathbf{u}_2$ the following way: for all $q, q' \in Q$, $\mathbf{u}(q, q') = 1$ if there exists $s \in Q$ such that $\mathbf{u}_1(q, s) = 1$ and $\mathbf{u}_2(s, q') = 1$ otherwise $\mathbf{u}(q, q') = 0$.

There is another unary operator on the elements of the monoid called the iteration operator and denoted by #. Intuitively it means that we repeat increasingly the words in the sequences abstracted by the limit word. Before we define this operator, we shall give the equivalent definitions for recurrence and idempotency of limit words.

Definition 2.15 (Idempotent limit words)**.** A limit-word $\mathbf{u}$ is called idempotent if

$$\mathbf{u} = \mathbf{u}^2.$$

Similarly for recurrence, except that we also ask that a recurrent limit word be necessarily idempotent.

Definition 2.16 (Recurrent states)**.** A state $r \in Q$ is called $\mathbf{u}$-recurrent for some limit word $\mathbf{u}$, if $\mathbf{u}$ is idempotent and for all $q \in Q$ the following holds:

$$\mathbf{u}(r, q) = 1 \implies \mathbf{u}(q, r) = 1.$$

Otherwise r is said to be $\mathbf{u}$-transient.

[5]The limit words are written with bold letters.
[6]The juxtaposition of two limit words denotes their concatenation.

The iteration unary operator removes the transitions that go to transient states, this relies on the fact that by repeating a word enough times we can make the probability distribution on the transient states be arbitrarily close to 0.

Definition 2.17 (Iteration of a limit word)**.** Given an idempotent limit word $\mathbf{u}$ we define the iteration of $\mathbf{u}$ denoted as $\mathbf{u}^{\#}$ in the following way: for all states $q, q' \in Q$, $\mathbf{u}^{\#}(q, q') = 1$ if $\mathbf{u}(q, q') = 1$ and q' is $\mathbf{u}$-recurrent otherwise $\mathbf{u}^{\#}(q, q') = 0$.

The identity element of this monoid is denoted by $\mathbf{1}$ and it is defined as: $\mathbf{1}(q, q) = 1$ for all $q \in Q$ and 0 otherwise. For every letter $a \in A$ we define the corresponding limit word $\mathbf{a}$ as follows: for all $q, q' \in Q$, $\mathbf{a}(q, q') = 1$ if $a(q, q') > 0$ and 0 otherwise.

Definition 2.18 (Markov monoid)**.** For a given probabilistic automaton $\mathcal{A}$ we call the Markov monoid of $\mathcal{A}$, denoted $\mathcal{G}$, the smallest set which is closed under concatenation and iteration operators and contains $\{\mathbf{1}\} \cup \{\mathbf{a} \mid \text{ for } a \in A\}$.

Note that this set has a finite cardinality, since they are just oriented graphs, it has at most $2^{|Q|^2}$ elements. Hence a procedure computing it would terminate. Computing the Markov monoid is exactly what the algorithm given further below does, and then searches for a value 1 witness. Before that, we motivate the definition of a value 1 witness with an example.

2.2.3 An example

Before moving further we remind the reader that the value 1 problem is the following: Given a PA $\mathcal{A}$ decide whether there exists a sequence of words $(u_n)_{n\in\mathbb{N}}$ such that for all $\epsilon > 0$ there exists an $n \in \mathbb{N}$ such that $1 - \mathcal{A}(u_n) < \epsilon$.

Example 2.4. *We provide now an example of an automaton with value 1.*

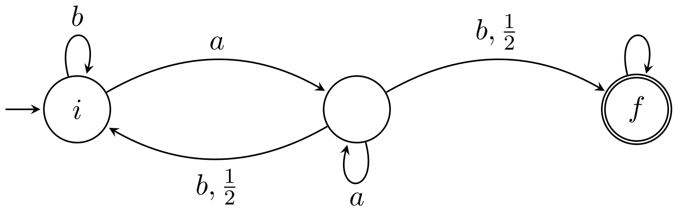

FIGURE 2.6: An automaton with value 1.

One can see that for this automaton, $ab(i, f) = \frac{1}{2}$, and $abab(i, f) = \frac{3}{4}$, and so on. Hence the sequence of words $((ab)^n)_{n\in\mathbb{N}}$ is the witness for the value 1 of this automaton.

Let us compute the Markov monoid associated to the automaton in Example 2.4. First the elements associated to the letters a and b respectively:

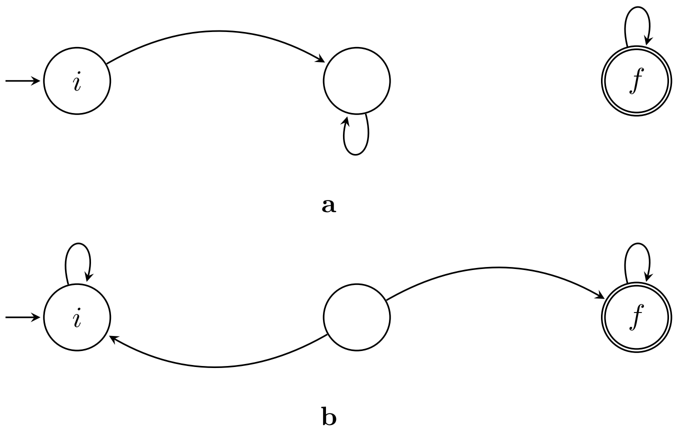

FIGURE 2.7: Elements of the Markov monoid of the PA in Example 2.4 associated to the letters a and b.

Further we see that $\mathbf{a} = \mathbf{aa}$ hence $\mathbf{a}$ is idempotent and we can iterate it, but since every state in $\mathbf{a}$ is recurrent except i we have $\mathbf{a}^{\#} = \mathbf{a}$. The same holds for $\mathbf{b}$, that is $\mathbf{b}$ is idempotent and since the only state in $\mathbf{b}$ that is transient is the one in the middle, and it does not have any incoming edges, we have $\mathbf{b}^{\#} = \mathbf{b}$.

Concatenating $\mathbf{a}$ and $\mathbf{b}$ in both ways we get two new elements of the Markov monoid.

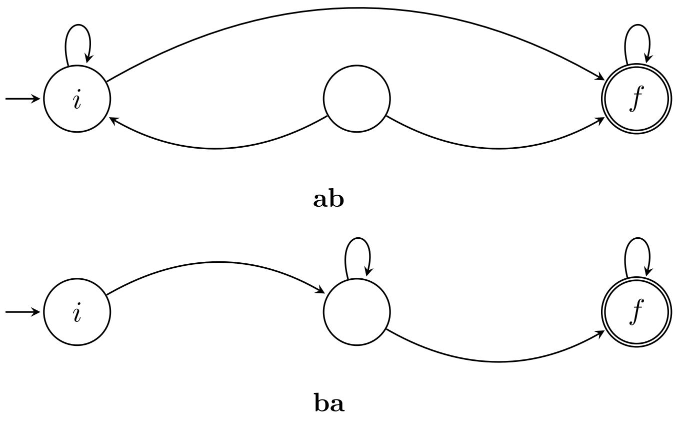

FIGURE 2.8: Elements of the Markov monoid of the PA in Example 2.4 associated to the words ab and ba.

Trying to concatenate these two limit words by themselves shows that they are idempotent. Before iterating them we identify that the state i is transient in $\mathbf{ab}$ because $\mathbf{ab}(i, f) = 1$ but $\mathbf{ab}(f, i) = 0$. Similarly for the middle state, thus we find that the only recurrent state in both limit words is f, we proceed by removing incoming edges to transient states.

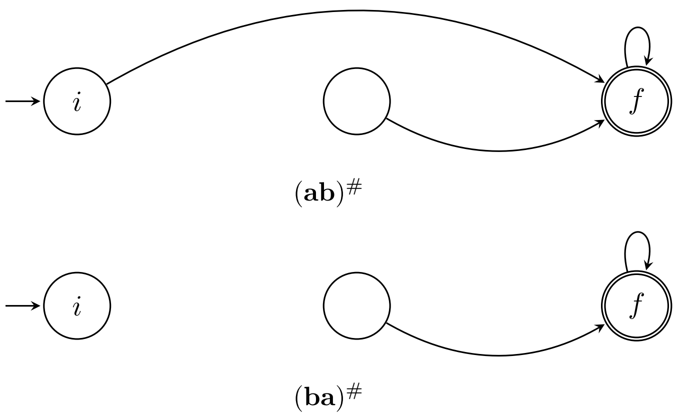

FIGURE 2.9: Elements of the Markov monoid of the PA in Example 2.4 associated to the iterations of ab and ba.

The difference between the limit word $(\mathbf{ab})^\#$ and the others is that $(\mathbf{ab})^\#$ is abstracting the asymptotic behavior where we go from the initial state to the final state and no other. This means that if $(\mathbf{ab})^\#$ reifies a sequence of words $(u_n)_{n\in\mathbb{N}}$ then the probability to accept words in this sequence will be arbitrarily close to 1. This gives us motivation for the definition in the following subsection.

2.2.4 Using the Markov monoid to decide value 1

Definition 2.19 (Value 1 witness)**.** Let $\mathcal{A}$ be a probabilistic automaton with initial state $i \in Q$, final states $F \subseteq Q$ and let $\mathcal{G}$ be its Markov monoid. A limit word $\mathbf{u} \in \mathcal{G}$ is called a value 1 witness if for all $q \in Q$,

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

Hence the first step of the algorithm is to compute the Markov monoid, and the last step is to look for a value 1 witness.

Algorithm 1 The Markov monoid algorithm.

```
Require: A probabilistic automaton.
1  $\mathcal{G} \leftarrow \{\mathbf{a} \mid a \in A\} \cup \{\mathbf{1}\}$.
2  repeat
3      if there is $\mathbf{u}, \mathbf{v} \in \mathcal{G}$ such that $\mathbf{u} \cdot \mathbf{v} \notin \mathcal{G}$ then
4          add $\mathbf{u} \cdot \mathbf{v}$ to $\mathcal{G}$
5      if there is $\mathbf{u} \in \mathcal{G}$ such that $\mathbf{u}$ is idempotent and $\mathbf{u}^\sharp \notin \mathcal{G}$ then
6          add $\mathbf{u}^\sharp$ to $\mathcal{G}$
7  until there is nothing to add
8  if there is a value 1 witness in $\mathcal{G}$ then
9      return true
10 else
11     return false
```

There are two properties that the Markov monoid must posses with respect to some automaton, in order for us to be sure, (non) existence of the value 1 witness corresponds to the (non) existence of the value 1 in the automaton: *consistency* and *completeness*.

We say that a sequence of words reifies a limit word, if the latter correctly abstracts the asymptotic behavior of the former:

Definition 2.20 (Reification)**.** A sequence of words $(u_n)_{n\in\mathbb{N}}$ reifies the limit word $\mathbf{u}$ if for all states $q, q' \in Q$ the sequence $(u_n(q,q'))_{n\in\mathbb{N}}$ converges and moreover:

$$\mathbf{u}(q,q') = 1 \iff \lim_{n\to\infty} u_n(q,q') > 0.$$

Definition 2.21 (Consistency)**.** A Markov monoid $\mathcal{G}$ is consistent with respect to an automaton $\mathcal{A}$ if for every limit word $\mathbf{u} \in \mathcal{G}$ there exists a sequence of words $(u_n)_{n\in\mathbb{N}}$ that reifies $\mathbf{u}$.

Definition 2.22 (Completeness)**.** A Markov monoid $\mathcal{G}$ is complete with respect to an automaton $\mathcal{A}$ if for every sequence of words $(u_n)_{n\in\mathbb{N}}$ there exists a limit word $\mathbf{u} \in \mathcal{G}$ such that for all states $q, q' \in Q$:

$$\limsup_{n\to\infty} u_n(q,q') = 0 \implies \mathbf{u}(q,q') = 0.$$

Every Markov monoid computed with Algorithm 1 is consistent, and if the automaton is leaktight it is also complete. If the Markov monoid is both consistent and complete, then the automaton has a value 1 if and only if there is a value 1 witness in the Markov monoid.

Theorem 2.23 ([FGO11]). *Let $\mathcal{G}$ be a consistent and complete Markov monoid with respect to an automaton $\mathcal{A}$, then $\mathcal{A}$ has a value 1 if and only if there exists a value 1 witness in $\mathcal{G}$.*

The Algorithm 1 terminates in less than $2^{|Q|^2}$ steps since at each step we are adding a limit word and there are less than $2^{|Q|^2}$ limit words. But this exponential upper bound can be improved to PSPACE by looking for a value 1 witness in a non-deterministic way and avoiding computation of the whole Markov monoid. Then using a similar argument as in [GO10] one can show that deciding the value 1 problem for leaktight automata is PSPACE-complete (the same is shown in [GO10] for the class of #-acyclic automata, which are a subclass of leaktight automata).

2.2.5 Consistency

Showing consistency is easier than showing completeness and it does not rely on the automaton being leaktight.

Lemma 2.24 ([FGO11]). *Let $(u_n)_{n\in\mathbb{N}}, (v_n)_{n\in\mathbb{N}}$ be two sequences of words that reify respectively the limit words $\mathbf{u}, \mathbf{v}$. Then the sequence of words $(u_n v_n)_{n\in\mathbb{N}}$ reifies the limit word $\mathbf{uv}$.*

Proof. We have that for all $q, q' \in Q$, $\mathbf{uv}(q,q') = 1$ implies that there exists $s \in Q$ such that $\mathbf{u}(q,s) = 1$ and $\mathbf{v}(s,q') = 1$. By the hypothesis $\lim_{n\to\infty} u_n(q,s) > 0$ and $\lim_{n\to\infty} v_n(s,q') > 0$, and from here $\lim_{n\to\infty} uv(q,q') > 0$. The other direction is almost symmetric. □

Lemma 2.25 ([FGO11]). *Let $(u_n)_{n\in\mathbb{N}}$ reify the limit word $\mathbf{u}$. Then there exists a map $f : \mathbb{N} \to \mathbb{N}$ such that $(u^n_{f(n)})_{n\in\mathbb{N}}$ reifies $\mathbf{u}^\#$.*

Proof. Since $(u_n)_{n\in\mathbb{N}}$ reifies $\mathbf{u}$, for all $q, q' \in Q$ there exists the limit $u(q,q') = \lim_{n\to\infty} u_n(q,q')$. Since $\mathbf{u}$ is idempotent[7] so must be u. The Markov chain with the transition matrix u is aperiodic[8] because of idempotency. For an aperiodic Markov chain with transition matrix u, there exists $\lim_{n\to\infty} u^n = v$ and for all transient states t and all states $q \in Q$, $(v)_{q,t} = 0$. Because matrix multiplication is continuous there exists a map $f : \mathbb{N} \to \mathbb{N}$

[7]Because $\mathbf{u}^\#$ is defined only for idempotent $\mathbf{u}$.

[8]The period of a state of a Markov chain is the greatest common divisor of the *return times* (how many steps it takes to come back to the state with positive probability, all $k \in \mathbb{N}$ such that $M^k(w,w) > 0$). A state that has period 1 is said to be aperiodic, and an automaton where every state is aperiodic is called aperiodic.

such that for all $n \in \mathbb{N}$:

$$||u^n - u^n_{f(n)}||_\infty = \max_{q,q' \in Q} |(u^n)_{q,q'} - (u^n_{f(n)})_{q,q'}| \leq \frac{1}{n}.$$

Hence the sequence $(v_n)_{n\in\mathbb{N}} = (u^n_{f(n)})_{n\in\mathbb{N}}$ converges to v. To show that $(v_n)_{n\in\mathbb{N}}$ reifies $\mathbf{u}^\#$ we proceed as follows. For all $q, q' \in Q$ we have:

$$\begin{array}{lrl}
\text{(by def. of iteration operator)} & \mathbf{u}^\#(q,q') = 1 & \iff q' \text{ is } \mathbf{u}\text{-recurrent and } \mathbf{u}(q,q') = 1 \\
\text{(because } u \text{ reifies } \mathbf{u}) & & \iff q' \text{ is } u\text{-recurrent and } u(q,q') > 0 \\
(u \text{ and } v \text{ have the same recurrence classes)} & & \iff q' \text{ is } v\text{-recurrent and } v(q,q') > 0 \\
\text{(no incoming edges to transient states)} & & \iff v(q,q') > 0 \\
\text{(from argument above)} & & \iff \lim\limits_{n\to\infty} v_n
\end{array}$$

□

Using the two lemmas above one can show by induction that the Markov monoid of some automaton, computed with Algorithm 1 is always consistent, and for every element $\mathbf{u}$ we can find a sequence of words which reifies $\mathbf{u}$.

Before we move on to sketch the completeness in the next section we provide this useful lemma. It essentially says that by repeating a limit word enough times we get an idempotent limit word.

Lemma 2.26 ([FGO11]). *For every limit word $\mathbf{u}$ the word $\mathbf{u}^{|Q|!}$ is idempotent.*

Proof. Assume that $\mathbf{u}^{|Q|!}(q,q') = 1$ then we have to show that $\mathbf{u}^{2|Q|!}(q,q') = 1$. Let $|Q|! = k$. Then since $k \geq |Q|$, we have $\mathbf{u}^k(q,q') = 1$ if and only if[9] there exists $r \in Q$ such that there exist $x, y, z \in \mathbb{N}$ fulfilling $x + y < |Q|$ and $x + y + z = k$ such that $\mathbf{u}(q,r)^x \mathbf{u}^y(r,r)\mathbf{u}^z(r,q') = 1$. Now we can repeat the second factor as many times as we want and since y divides $k = |Q|!$ we can repeat it i times such that $x + iy + z = 2k$. □

2.2.6 Leak witness

As it turns out, the question, whether some probabilistic automaton is leaktight or not, is decidable. And it can be decided by searching for a *leak-witness*.

[9]Since there is a finite number of states, if we look at $\mathbf{u}^k$ as a path of length k in the graph given by $\mathbf{u}$, and since $k > |Q|$ we must see a state r more than once, even before we reach the first $|Q|$ steps of this path.

But first we need a small generalization of the Markov monoid, namely we want to remember which transitions have been deleted by taking iterations. To accomplish this we take pairs of limit words, where the first component is as before, but when we take iteration of the second component we do not remove transitions to transient states.

Definition 2.27 (Extended limit word)**.** An extended limit word is a couple of limit words, denoted $(\mathbf{u}, \mathbf{u}_+)$, and an extended Markov monoid (usually denoted $\mathcal{G}_+$) is a monoid with the operations defined below.

The concatenation of two limit words is component-wise:

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

And in the case of iteration we only take the iteration of the first component (remembering the transition that was removed in the second component):

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

and it is defined only for an idempotent extended limit word.

As in Algorithm 1 we can compute this extended Markov monoid by taking concatenations and iterations of the elements of the set $\{(\mathbf{a}, \mathbf{a}) \mid \text{for } a \in A\}$.

The necessary and sufficient condition for an automaton to be leaktight is for it to have a certain element in the extended Markov monoid known as *leak witness*.

Definition 2.28. (Leak witness) An extended limit word $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{G}_+$ is called a leak witness if there exist states $r, t \in Q$ such that:

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

Intuitively we can see that these three conditions are enforcing the last three conditions of a leak in 2.12, respectively. And indeed such a connection exists:

Lemma 2.29 ([FGO11])**.** *A probabilistic automaton $\mathcal{A}$ has a leak if and only if there exists a leak witness in its extended Markov monoid.*

2.2.7 Completeness

The completeness relies in an algebraic argument about the existence of #-factorization trees of bounded height, see: [Sim90], [Sim94], [Col09], [Tor11].

We remind the reader that by $p_{\min}$ we denoted the smallest probability of a transition in an automaton that is the smallest non-zero element in the matrices M_a for $a \in A$.

The main lemma (completeness is just a corollary of this lemma) is called the *lower bound lemma* and says that for any finite word we can find some element $\mathbf{u}$ of the Markov monoid such that if with this limit word we can go from state q to state q', then so can we with the finite word and moreover the probability to go from q to q' with the finite word has a lower bound.

Lemma 2.30 ([FGO11])**.** *(Lower bound lemma) Let $\mathcal{A}$ be a leaktight automaton then for every finite word $u \in A^*$ there exists an extended limit word $(\mathbf{u}, \mathbf{u}_+)$ element of the extended Markov monoid such that for all states $q, q' \in Q$ we have:*

$$\mathbf{u}_+(q,q') = 1 \iff u(q,q') > 0 \text{ ,and}$$

$$\mathbf{u}(q,q') = 1 \implies u(q,q') > p_{\min}^{2^{3\cdot 2^{2\cdot|Q|^2}}} .$$

From Lemma 2.30 the completeness comes as a consequence of the extended Markov monoid being finite.

Lemma 2.30 itself is proved inductively on the height of the #-factorization tree for the given word. From the work cited above we know that there exists such a factorization tree with bounded height, a bound which does not depend on the length of the word.

For some monoid M let $E(M)$ be the set of idempotent words.

Definition 2.31 (Stabilization monoid)**.** A stabilization monoid $(M, \cdot, \sharp)$ is a finite monoid $(M, \cdot)$ equipped with an iteration operation $\sharp : E(M) \to E(M)$ such that:

$$\begin{aligned}
&(a \cdot b)^\sharp \cdot a = a \cdot (b \cdot a)^\sharp \qquad \text{for } a \cdot b \in E(M) \text{ and } b \cdot a \in E(M) \ , \\
&(e^\sharp)^\sharp = e^\sharp \qquad \text{for } e \in E(M) \ , \\
&e^\sharp \cdot e = e^\sharp \qquad \text{for } e \in E(M) \ .
\end{aligned}$$

It can be shown that the Markov monoid is a stabilization monoid.

Definition 2.32. Let A be a finite alphabet, $(M, \cdot, \sharp)$ a stabilization monoid and $\phi : A^* \to M$ a morphism. A $\sharp$-*factorization 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 labeled by $(u, \mathbf{u})$, for some $\mathbf{u} \in M$,

ii) every internal node with two children labeled by $(u_1, \mathbf{u}_1)$ and $(u_2, \mathbf{u}_2)$ is labeled by $(u_1 \cdot u_2, \mathbf{u}_1 \cdot \mathbf{u}_2)$,

iii) every leaf is labeled by $(a, \mathbf{a})$ where a is a letter.

iv) every internal node with three or more children is labeled by $(u_1 \dots u_n, \mathbf{e}^{\#})$ for some $\mathbf{e} \in E(M)$, and its children are labeled 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.

And we can give a higher bound on the depth of these #-factorization trees that does not depend on the length of the word.

Theorem 2.33 ([Sim90, Sim94, Col09, Tor11])**.** *Let A be a finite alphabet, $(M, \cdot, \sharp)$ a stabilization monoid and $\phi : A^* \to M$ a morphism. Every word $u \in A^*$ has a $\sharp$-factorization tree whose depth is less than $3 \cdot |M|$.*

Chapter 3

Comparison of Simple and Leaktight Automata

In this chapter we prove that the class of simple automata is a strict subset of the class of leaktight automata. We proceed by first showing inclusion and then giving an example of an automaton which is not simple but is leaktight.

3.1 Inclusion

First we introduce the notion of a *non-simplicity witness*, and prove that if the extended Markov monoid contains such a witness, then we can build a non-simple process, making the automaton non-simple. After this we show that the existence of a leak witness in the Markov monoid implies the existence of the non-simplicity witness finalizing the inclusion.

Definition 3.1 (Non-simplicity witness)**.** Call the triple $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ of elements of the Markov monoid of an automaton $\mathcal{A}$ a *non-simplicity* witness if there exist states $r, t \in Q$ such that:

1. $\mathbf{y}$ and $\mathbf{x}\mathbf{y}^{\#}\mathbf{z}$ are idempotent,
2. r is $\mathbf{x}\mathbf{y}^{\#}\mathbf{z}$-recurrent,
3. $\mathbf{x}\mathbf{y}(r, t) = 1$, and
4. t is $\mathbf{y}$-transient.

Note that (1) is implied by (2).

For instance in the Example 2.2, by computing a part of the Markov monoid we get a non-simplicity witness: $(\mathbf{b}, \mathbf{a}, \mathbf{1})$ for states r and t.

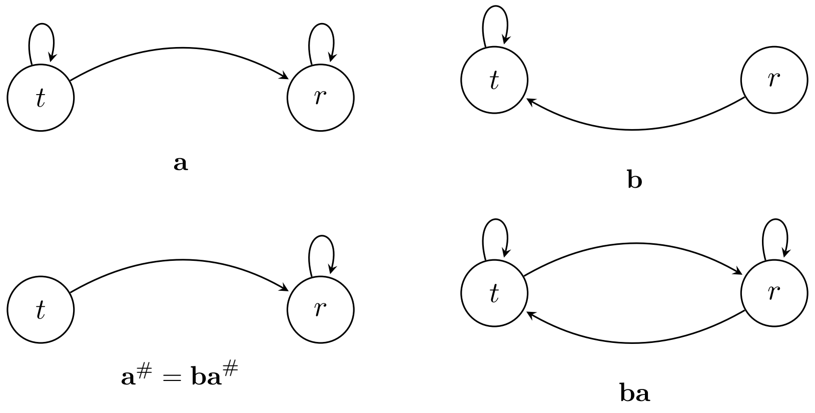

FIGURE 3.1: Elements of the Markov monoid of Example 2.2

We see that t is $\mathbf{a}$-transient, because $\mathbf{a}(t, r) = 1$ but $\mathbf{a}(r, t) = 0$. And we have that r is $\mathbf{ba}^{\#}$-recurrent and that $\mathbf{ba}(r, t) = 1$. Hence in this example the triple $(\mathbf{b}, \mathbf{a}, \mathbf{1})$ with the respective states r,t is a non-simplicity witness.

In order for us to show that non-simplicity witness implies that the automaton is not simple we use the witness to build an infinite word and then use Lemma 2.8. Before that we give the following Lemma which helps us fulfill the first hypothesis of Lemma 2.8.

Lemma 3.2. *Let $\mathbf{u}$ be an element of the Markov monoid, $r \in Q$ that is $\mathbf{u}$-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 $f : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$

$$u_{f(1)} \dots u_{f(n)}(r, r) > \gamma.$$

Proof. From the definition of reification we have that the sequence of words $(u_n)_{n \in \mathbb{N}}$ reifies a limit-word $\mathbf{u}$ if for all $q, q' \in Q$, $\lim_n u_n(q, q')$ exists and

$$\lim_n u_n(q, q') > 0 \iff \mathbf{u}(q, q') = 1. \tag{3.1}$$

As a consequence of (3.1) there exists a strictly increasing map $f : \mathbb{N} \to \mathbb{N}$ such that the following two conditions hold:

1. for all $n \in \mathbb{N}$,
$$l_n = \sum_{q, q' \in Q \wedge \mathbf{u}(q, q') = 0} u_{f(n)}(q, q') < \frac{1}{2^{n+c}} \tag{3.2}$$

holds for a constant $c \in \mathbb{N}$

2. there exists $d > 0$ such that for all $q, q' \in Q$ and $n \in \mathbb{N}$:

$$\mathbf{u}(q, q') = 1 \implies u_{f(n)}(q, q') > d. \tag{3.3}$$

Define R to be the set

$$R = \{q \in Q \mid \mathbf{u}(r, q) = 1\}$$

From idempotency and recurrence, for all $q \in R$ and $q' \in Q$

$$\mathbf{u}(q, q') = 1 \implies q' \in R \tag{3.4}$$

Because on the contrary if for some $q \in R$ we have $\mathbf{u}(q, q') = 1$ and $q' \notin R$ then by definition of R we have that $\mathbf{u}^2(r, q') = 1$ and because of idempotency $\mathbf{u}(r, q') = 1$ but then $q' \in R$.

Let $w_n = u_{f(1)} u_{f(2)} \dots u_{f(n)}$, then we have

$$\begin{aligned} w_n(r, R) = 1 - w_n(r, Q \setminus R) &= 1 - \sum_{q \in Q \setminus R} w_n(r, q) \\ &= 1 - \sum_{\mathbf{u}(r,q)=0} w_n(r, q) \\ &\geq 1 - \frac{1}{2^{n+c}} \geq 1 - \frac{1}{2^c} \end{aligned}$$

Where the last equality is from contraposition of (3.4) and the inequality comes from (3.2).

Since r is $\mathbf{u}$-recurrent, for all $q \in R$ we have $\mathbf{u}(q, r) = 1$, and using (3.3) we get that $u_{f(n+1)}(q, r) > d$. From this bound and the equation above, it follows that:

$$\begin{aligned} w_{n+1}(r, r) &\geq \sum_{q \in R} w_n(r, q) u_{f(n+1)}(q, r) \\ &\geq \sum_{q \in R} w_n(r, q) d \\ &\geq d(1 - \frac{1}{2^c}), \end{aligned}$$

which completes the proof for $\gamma = d(1 - \frac{1}{2^c})$. □

Now we are ready to state and prove that existence of a non-simplicity witness is a sufficient condition for the automaton to be non-simple.

Lemma 3.3. *Let $\mathcal{A}$ be a PA. If the Markov monoid of $\mathcal{A}$ contains a non-simplicity witness, then $\mathcal{A}$ is not simple.*

Proof. Let $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be the non-simplicity witness, define $\mathbf{u} = \mathbf{x}\mathbf{y}^{\#}\mathbf{z}$, and $r, t \in Q$ such that r is $\mathbf{u}$-recurrent, t is $\mathbf{y}$-transient, and $\mathbf{xy}(r, t) = 1$.

Let $(x_n)_{n\in\mathbb{N}}, (y_n)_{n\in\mathbb{N}}, (z_n)_{n\in\mathbb{N}}$ be sequences of words which reify $\mathbf{x}, \mathbf{y}, \mathbf{z}$ respectively. These sequences exist for any automaton because of the consistency of the Markov monoid.

We are going to transform these sequences of words into other sequences while keeping the reification, in order to gain some additional properties (in particular (3.5) and (3.6)) which are useful in the sequel.

According to Lemmas 2.24 and 2.25 there exists a map g such that $(x_n y^n_{g(n)})_{n\in\mathbb{N}}$ reifies $\mathbf{x}\mathbf{y}^{\#}$ and $(x_n y^n_{g(n)} z_n)_{n\in\mathbb{N}}$ reifies $\mathbf{u}$.

From the definition of a non-simplicity witness we have $\mathbf{xy}(r, t) = 1$ and $\mathbf{y}$ is idempotent. Idempotency of $\mathbf{y}$ implies $\mathbf{xy}(r, t) = \mathbf{x}\mathbf{y}^k(r, t) = 1$ for all $k \in \mathbb{N}$. Now, from Lemma 2.24 for some fixed $k \in \mathbb{N}$,

$$x_n y^k_{g(n)} \text{ reifies } \mathbf{x}\mathbf{y}^k,$$

from which: there exists some $M_k \in \mathbb{N}$ such that for all $n \geq M_k$, $x_n y^k_{g(n)}(r, t) > 0$. This way we construct another increasing sequence of integers: $(M_n)_{n\in\mathbb{N}}$. In order to simplify the notation define: $h(n) = n + M_n$ and $h'(n) = g(n + M_n)$.

Since $h : \mathbb{N} \to \mathbb{N}$ is increasing and by definition of reification we have:

$$x_{h(n)} y^n_{h'(n)} z_{h(n)} \text{ reifies } \mathbf{x}\mathbf{y}^{\#}\mathbf{z},$$

and also by choice of the function h

$$\forall n \in \mathbb{N}, x_{h(n)} y^n_{h'(n)}(r, t) > 0. \tag{3.5}$$

Let f be the strictly increasing map from applying Lemma 3.2 to the limit word $\mathbf{u}$, state r and sequence of words $(x_{h(n)} y^n_{h'(n)} z_{h(n)})_{n\in\mathbb{N}}$. Define the new sequence of words $(U_n)_{n\in\mathbb{N}} = (X_n Y_n Z_n)_{n\in\mathbb{N}}$ such that $X_n = x_{h(f(n))}$, $Y_n = y^{f(n)}_{h'(f(n))}$, $Z_n = z_{h(f(n))}$.

Then for some $\gamma > 0$ we have:

$$U_1 \ldots U_n(r, r) > \gamma, \tag{3.6}$$

for all $n \in \mathbb{N}$.

Let $\rho = U_1U_2\dots$ and $W_n = U_1U_2\dots U_n$. We are going to prove that the word ρ satisfies the hypothesis of Lemma 2.8. That is we want to prove the following, for the word ρ, initial distribution α corresponding to state r, and states r and t:

1. there exists γ such that for infinitely many $n \in \mathbb{N}$, $\rho_{\leq n}(r,r) > \gamma$,
2. for some $k_n \geq 1$, $\rho[n+1, n+k_n](r,t) > 0$, and
3. $\liminf_n \rho_{\leq n+k_n}(r,t) = 0$.

The proof of these three statements is the following:

1. immediate from (3.6). We choose the sequence $(n_i)_{i\in\mathbb{N}}$ for which $\rho_{\leq n_i}(r,r) > \gamma$ holds as $n_i = \sum_{k=1}^{i} |U_k|$.
2. since (3.5) holds for all $n \in \mathbb{N}$, it must hold also for $f(n), n \in \mathbb{N}$. We choose
$$k_i = |X_iY_i|.$$
3. by the choice of $(k_n)_{n\in\mathbb{N}}$ above, we have that $\rho[n+1, n+k_n] = X_nY_n$ which reifies $\mathbf{xy}^{\#}$. And t is $\mathbf{y}$-transient by choice hence for all $q \in Q$, $\mathbf{xy}^{\#}(q,t) = 0$. From reification $\lim_n X_nY_n(q,t) = 0$. Now,
$$\rho_{\leq n+k_n}(r,t) = \sum_{q\in Q} \rho_{\leq n}(r,q)\rho[n+1,n+k_n](q,t) = \sum_{q\in Q} \rho_{\leq n}(r,q)X_nY_n(q,t),$$
and the second factor converges to zero.

Then by Lemma 2.8, ρ induces a non-simple process in $\mathcal{A}$ making it non-simple. □

The next Lemma states that the existence of a leak witness in the Markov monoid, implies the existence of a non-simplicity witness. Before that we need one more definition.

Definition 3.4 (#-height)**.** Every extended limit word $(\mathbf{u}, \mathbf{u}_+)$ that is a part of some extended Markov monoid can be decomposed (not uniquely) into expressions with concatenation binary operator, iteration unary operator, and $\{(\mathbf{a}\,\mathbf{a}) |\ a \in A\} \cup \{(\mathbf{1}, \mathbf{1})\}$ as operands. Then $\# - \text{height}(\mathbf{u})$ is the minimum out of all its decompositions, of the deepest nesting of iterations. Given some decomposition $\mathbf{u} = \mathbf{v}(\mathbf{v}_1(\mathbf{v}_2(\dots(\mathbf{v}_n)^{\#}\dots)^{\#}\dots)^{\#})\dots$ we say that $\# - \text{height}(\mathbf{u}) = n$ if there are no deeper nestings of the iteration operator.

Lemma 3.5. *Let $\mathcal{A}$ be a PA that contains a leak, then it also contains a non-simplicity witness.*

Proof. Let $\mathcal{C} \subseteq \mathcal{G}_+$ (where $\mathcal{G}_+$ is the extended Markov monoid) be the set of all pairs $(\mathbf{u}, \mathbf{u}_+)$ of the extended Markov monoid for which there exist states $r, t \in Q$ such that

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

We see that $\mathcal{C} \neq \emptyset$ because of the following: let $(\mathbf{u}, \mathbf{u}_+)$ be a leak witness, then there exist states $r, t \in Q$ such that r is $\mathbf{u}$-recurrent, $\mathbf{u}_+(r, t) = 1$ and $\mathbf{u}(t, r) = 0$. From recurrence of r we have that $\mathbf{u}(r, t) = 0$, therefore $(\mathbf{u}, \mathbf{u}_+) \in \mathcal{C}$.

Let $(\mathbf{w}, \mathbf{w}_+) \in \mathcal{C}$ have a decomposition into iteration and concatenation with the smallest #-height in the set, i.e for all $(\mathbf{u}, \mathbf{u}^+) \in \mathcal{C}$, $\#\text{-height}((\mathbf{w}, \mathbf{w}_+)) \leq \#\text{-height}((\mathbf{u}, \mathbf{u}_+))$. And let $r, t \in Q$ such that $\mathbf{w}_+(r, t) = 1$, $\mathbf{w}(r, t) = 0$ and r is $\mathbf{w}$-recurrent.

Because $\mathbf{w} \neq \mathbf{w}_+$ in any decomposition of this extended limit-word into concatenation and iteration of letters there must be at least one iteration operator hence we can write $\mathbf{w} = \mathbf{x}\mathbf{y}^{\#}\mathbf{z}$ for some $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{G}$ such that $\mathbf{xyz}$ has a strictly smaller #-height than $\mathbf{w}$. Define these two sets:

$$R = \{q \in Q \mid \mathbf{w}(r, q) = 1\}$$
$$T = \{q \in Q \mid q \text{ is } \mathbf{y}\text{-transient}\} .$$

It is an easy exercise to show, using idempotency and recurrence of r, that every state in R is also $\mathbf{w}$-recurrent[1]. Now we claim and prove the following.

$$\text{There exist } r' \in R \text{ and } t' \in T \text{ such that } \mathbf{xy}(r', t') = 1. \tag{3.7}$$

The proof of (3.7) is by contradiction, showing that if it does not hold, then $(\mathbf{xyz}, \mathbf{w}_+) \in \mathcal{C}$ and $\mathbf{xyz}$ has a strictly smaller #-height then $\mathbf{x}\mathbf{y}^{\#}\mathbf{z}$.

Suppose on the contrary that for all $r' \in R$ and $t' \in T$, $\mathbf{xy}(r', t') = 0$ then for all $q \in Q$ we have $\mathbf{xy}(r', q) = \mathbf{x}\mathbf{y}^{\#}(r', q)$ because the iteration operator deletes only the incoming

[1]Also essentially the same set is defined in the proof of Lemma 3.2 and recurrence is implied from there.

transitions to states that are $\mathbf{y}$-transient. And from here

$$\forall r' \in R, \forall q \in Q,\ \mathbf{xyz}(r', q) = \mathbf{xy}^{\#}\mathbf{z}(r', q), \tag{3.8}$$

also from idempotency of $\mathbf{w}$ we have that for all $r' \in R$ and $q \in Q$ $\mathbf{w}^k(r', q) = \mathbf{w}(r', q) = 1$ implies that $q \in R$. But $\mathbf{w}' = \mathbf{xyz}$ has strictly smaller $\#$-height than $\mathbf{w} = \mathbf{xy}^{\#}\mathbf{z}$, and $(\mathbf{w}', \mathbf{w}_+)^{|Q|!} \in \mathcal{C}$ because all of the conditions of $\mathcal{C}$ are satisfied by $(\mathbf{w}', \mathbf{w}_+)^{|Q|!}$ and states r and t:

1. $\mathbf{w}'^{|Q|!}$ is idempotent,
2. r is $\mathbf{w}'^{|Q|!}$-recurrent,
3. $\mathbf{w}'^{|Q|!}(r, t) = 0$, and
4. $\mathbf{w}_+^{|Q|!}(r, t) = 1$.

The proof is the following:

1. immediate from Lemma 2.26,
2. let $q \in Q$ such that $\mathbf{w}'^{|Q|!}(r, q) = 1$. Since r is $\mathbf{w}$-recurrent and $\mathbf{w}$ is idempotent, $\mathbf{w}(r, r) = 1$ and $r \in R$. Then by (3.8) $\mathbf{w}^{|Q|!}(r, q) = 1$. Since r is $\mathbf{w}$-recurrent $\mathbf{w}(q, r) = 1$ and $q \in R$, by (3.8), $\mathbf{w}(q, r) = \mathbf{w}'(q, r)$ and $\mathbf{w}'(q, r) = 1$. By the same argument $\mathbf{w}'(r, r) = 1$ thus $\mathbf{w}'^{|Q|!}(q, r) = 1$.
3. since $\mathbf{w}(r, t) = 0$, and from idempotency of $\mathbf{w}$ we have $\mathbf{w}^2(r, t) = 0$, applying (3.8), we have $\mathbf{w}'(r, t) = \mathbf{w}'^2(r, t) = 0$ and from here $\mathbf{w}'^{|Q|!}(r, t) = 0$.
4. immediate from idempotency of $\mathbf{w}_+$.

Thus $(\mathbf{w}', \mathbf{w}_+)^{|Q|!} \in \mathcal{C}$ and has strictly smaller $\#$-height, a contradiction. This concludes the proof of (3.7).

Note that (3.7) is sufficient for $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ to be a non-simplicity witness with the states r' and t' chosen from (3.7), because of the following:

1. since $(\mathbf{w}, \mathbf{w}_+) \in \mathcal{C}$, $\mathbf{w}$ must be idempotent, and since we can only take the interation of idempotent limit words, by construction $\mathbf{y}$ is idempotent,
2. from the remark above about the set R, since $r' \in R$ it is $\mathbf{w}$-recurrent,
3. $\mathbf{xy}(r', t') = 1$ from (3.7).

4. t' is $\mathbf{y}$-transient, from definition of T, and since $t' \in T$ in (3.7).

□

Theorem 3.6. *Leaktight automata contain simple automata.*

Proof. From Lemma 3.5 every automaton $\mathcal{A}$ that contains a leak also contains a non-simplicity witness, and from Lemma 3.3 such an $\mathcal{A}$ is not simple. □

3.2 Strict inclusion

We prove the strict inclusion by providing an example of an automaton that is not simple but is leaktight.

Consider the automaton given in Example 2.2 which we give below and the initial distribution corresponding to the initial state.

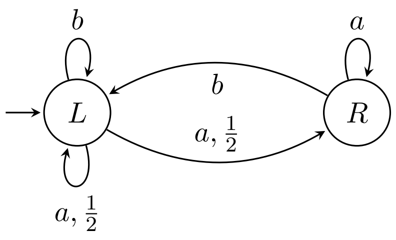

FIGURE 3.2: A nonsimple automaton $\mathring{A}$

Let $u_n = a^n b$, we look at the word $w = u_1 u_2 \ldots$. Let $v_n = u_1 u_2 \ldots u_n$. We use Lemma 2.8, we have that for all $n \in \mathbb{N}$, $v_n(L, L) = 1$, fulfilling the first hypothesis. And we also have that for all $n \in \mathbb{N}$, $v_n a^{n+1}(L, L) > 0$, fulfilling the second hypothesis. And finally we have that

$$v_n a^{n+1}(L, L) = \frac{1}{2^{n+1}}$$

which cannot be bounded from below by any $\gamma > 0$.

But by computing the extended Markov monoid of the automaton above we can see that it has no leak witness making it leaktight.

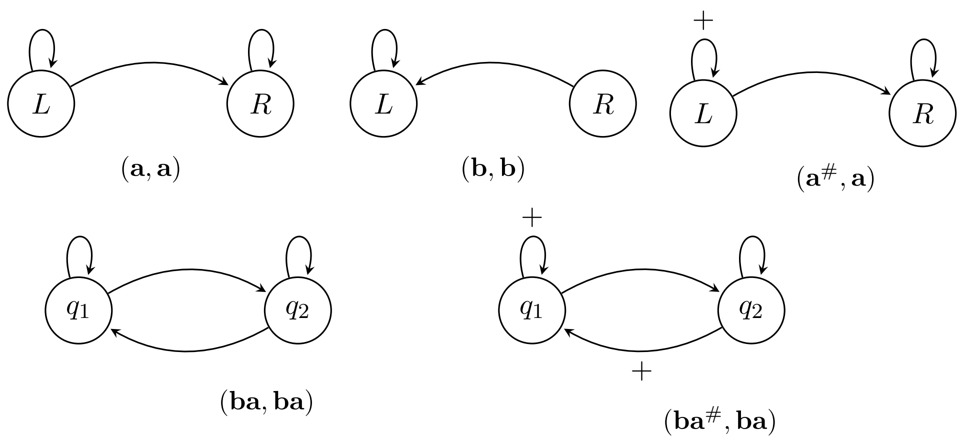
+
L
R
$(\mathbf{a},\mathbf{a})$
$(\mathbf{b},\mathbf{b})$
$(\mathbf{a}^{\#},\mathbf{a})$
q_1
q_2
$(\mathbf{ba},\mathbf{ba})$
$(\mathbf{ba}^{\#},\mathbf{ba})$

FIGURE 3.3: Extended Markov monoid of $\mathcal{A}$

The transition marked with "+" in Figure 3.3 are transitions which exists in the left component, but not the right one.[2]

The monoid consists of the five elements in Figure 3.3, and the following relations hold:

$$\begin{array}{llll} \mathbf{a}^2 = a & \mathbf{b}^2 = \mathbf{b} & \mathbf{ab} = \mathbf{b} & \mathbf{aa}^{\#} = \mathbf{a}^{\#} \\ \mathbf{a}^{\#2} = \mathbf{a}^{\#} & (\mathbf{ba}^{\#})^2 = \mathbf{ba}^{\#} & \mathbf{a}^{\#}\mathbf{a} = \mathbf{a}^{\#} & \mathbf{a}^{\#}\mathbf{b} = \mathbf{b} \end{array} .$$

But none of the elements in Figure 3.3 is a leak witness.

Since $\mathcal{A}$ in Figure 3.3 is leaktight but not simple and according to Theorem 3.6 leaktight automata include simple automata, we conclude that the inclusion is strict.

[2]For example the loop on L exists in $\mathbf{a}$ but not in $\mathbf{a}^{\#}$.

Chapter 4

Quantum automata

Quantum automata are the quantum analog of probabilistic automata. The current state of the automaton is given as a *superposition* of the set of states. A few definitions of such automata have been given in the literature, notably *measure many* quantum automata in [KW97] and *measure once* quantum automata given in [MC00]. Quantum automata are strictly weaker than probabilistic automata. We are interested in them because of the positive decidability results in [DJK05] and [BJKP05]. Here they show that the strict emptiness problem for measure once automata is decidable. We hope to use the same techniques to enlarge the class of PAs with decidable value 1 problem.

Problems about measure many quantum automata are mostly undecidable, like probabilistic ones. Therefore we are mostly concerned with measure once quantum automata and in the sequel this is what we call a quantum automaton.

4.1 Definitions

Definition 4.1 (Quantum automaton)**.** A quantum automaton $\mathcal{QA}$ is the 5 tuple $\mathcal{QA} = (Q, A, i, (U_a)_{a\in A}, P)$ where:

- Q is a finite set of states with cardinality n,
- A is a finite alphabet,
- i is an initial state,
- $(U_a)_{a\in A}$ are a set of unitary[1] matrices, elements of $\mathbb{C}^{n\times n}$,

[1]A unitary matrix A is characterized by the equation: $AA^* = I$ where A^* is the conjugate transpose of A.

- P is an $n \times n$ orthogonal projection[2].

The quantum automaton maps words $w \in A^*$ to reals in $[0,1]$. And this mapping function is given by:

$$\mathcal{QA}(w) = ||\mathbf{i}U_w P||^2.$$

Where with $\mathbf{i}$ we write the $1 \times n$ row matrix that has 1 in the place of the state $i \in Q$ and 0 everywhere else. Given some vector $\mathbf{v}$, with $||\mathbf{v}||$ we write the euclidean vector norm. And with U_w we write $U_{a_1}U_{a_2}\ldots U_{a_n}$ where $w = a_1a_2\ldots a_n$.

Before we show that quantum automata are weaker than the probabilistic ones we are going to define the generalized probabilistic automaton and tensor product of vectors.

Definition 4.2 (Generalized probabilistic automaton)**.** A generalized PA is a PA where we relax the condition of the transition matrices to be stochastic. That is the vectors $\mathbf{i}$, $\mathbf{F}$ and the matrices $(M_a)_{a\in A}$ are arbitrary real numbers.

It turns out that this relaxation does not give any more expressive power:

Theorem 4.3 ([Tur69])**.** *Let L be a language that is recognized by a generalized PA with cutpoint λ ($L = \mathcal{L}_>(\mathcal{QA}, \lambda)$). Then there exists a PA $\mathcal{A}$ and cutpoint λ' such that $L = \mathcal{L}_>(\mathcal{A}, \lambda')$.*

Definition 4.4 (Tensor product)**.** Let u and v be vectors of dimension m and n respectively. Their tensor product $u \otimes v$ is the vector with dimension mn such that $(u\otimes v)_{<i,j>} = u_i v_j$ where $< i,j >= n(i-1)+j$. And if M and N are $m\times m$ and $n\times n$ matrices respectively then $M\otimes N$ is the $mn\times mn$ matrix with $(M\otimes N)_{<i,k>,<j,l>} = M_{i,j}N_{k,l}$.

We would like to turn the function $\mathcal{QA}(w) = ||\mathbf{i}U_w P||^2$ into some function $\mathbf{i}M_w\mathbf{F}$ where all these matrices have real elements, we call this transformation a *bilinearization.* Then from the previous theorem we know that for a language defined by such a function, there exists a PA which recognizes it.

Bilinearization into a language that is recognized by some generalized PA relies on the fact that a complex number $x + iy$ can be identified by the matrix $\begin{pmatrix} x & -y \\ y & x \end{pmatrix}$. Doubling matrices like this, maps unitary matrices to orthogonal matrices[3], and complex matrices of orthogonal projection to real matrices of orthogonal projection[4].

[2]A complex matrix of orthogonal projection is characterized by the equation: $A = A^* = A^2$.

[3]An orthogonal matrix is characterized by the equation $AA^T = I$ where A^T is the transpose of the matrix A.

[4]Real matrices of orthogonal projection are characterized by the equation: $A = A^T = A^2$.

Theorem 4.5 ([MC00])**.** *Let L be a language that is recognized by some quantum automaton. Then there exists a generalized PA that recognizes L.*

Proof. (Sketch.) Let $\mathcal{QA}$ be the quantum automaton that recognizes L and let p_i be the i-th column of P. Then for some word $w \in A^*$ we have:

$$\begin{aligned}\mathcal{QA}(w) &= \sum_{i=0}^{n} ||\mathbf{i}U_w p_i||^2 \\ &= \sum_{i=0}^{n} \mathbf{i}(U_w^* \otimes U_w)(p_i^* \otimes p_i) \\ &= \mathbf{i}(U_w^* \otimes U_w)(\sum_{i=0}^{n} p_i^* \otimes p_i)\end{aligned}$$

Since the matrices above still might have complex elements, we simulate them using 2×2 matrices, thus getting a generalized PA of dimension $2n^2$. □

4.2 Decidability results

The reason why we are interested in quantum automata are these decidability results:

Problem 4.1. *Let $\mathcal{QA}$ be a quantum automaton and λ a cutpoint. Define the language L as $L = \mathcal{L}_\diamond(\mathcal{QA}, \lambda) = \{w \in A^* \mid \mathcal{QA}(w) \diamond \lambda\}$. Where $\diamond \in \{\geq, \leq, >, <, =\}$.*

The curious result is:

Theorem 4.6 ([BJKP05],[DJK05])**.** *Problem 4.1 is undecidable when $\diamond \in \{\geq, \leq, =\}$ but it is decidable when $\diamond \in \{>, <\}$.*

And more related to the work in this thesis:

Problem 4.2. *Given a quantum automaton $\mathcal{QA}$, and cutpoint λ, decide whether λ is isolated or not.*

Theorem 4.7 ([BJKP05],[DJK05])**.** *Problem 4.2 is decidable.*

One might ask for intuition as to why such similar problems as the emptiness for $\geq$ and emptiness for $>$ are not both decidable or both undecidable, and the answer is the following. In order to decide emptiness for $>$ we work on the group generated by the transition matrices and then make it compact by adding all the limit points. This last transformation loses the information that is necessary for deciding equality. We will provide a few more details in the following.

The undecidability result is a reduction from Post's correspondence problem to the emptiness for $\geq$. Where for the reduction the words are encoded by two orthogonal matrices that represent the rotations of angle $\arccos(\frac{3}{5})$ on the first and third axes. See [BJKP05] for more details.

The decidability result is more interesting and it relies on the following ideas.

Definition 4.8 (Compact group)**.** A group of real matrices G is called compact if for every element $A \in G$ if $\lim_n A^n$ exists then it is in G.

Definition 4.9 (Algebraic group)**.** We call a group of $n \times n$ real matrices G, algebraic, if and only if there exists a finite subset P of $\mathbb{C}[x_{1,1}, \dots, x_{n,n}]$ such that $X \in G$ if and only if X is the common zero of the polynomials in P.

First main idea that leads to decidability comes from algebraic geometry:

Theorem 4.10 ([OVL90])**.** *A compact group of real matrices is algebraic.*

Taking Theorem 4.10 for granted we give a very general overview of the proof of Theorem 4.6.

We look at the group G that is generated by the transition matrices of the quantum automaton (after the simulation of the complex numbers by 2×2 matrices). We take the Euclidean closure[5] of $\bar{G}$ of G. Now $\bar{G}$ is a group, because G is a group generated by orthogonal matrices for which the following is true: for all $X \in G$ there exists a sequence of natural numbers $(k_n)_{n \in \mathbb{N}}$ such that $\lim_n X^{k_n} = I$. Thinking about the geometrical interpretation of orthogonal matrices, that is, that they correspond to rotations or reflections, it is obvious that by doing them enough times we come back to the identity transformation. In the case where the rotation is an irrational factor of π we need to take the limit.

From Theorem 4.10 we have that there exists $P \in \mathbb{C}[x_{1,1}, \dots, x_{n,n}]$ such that $X \in \bar{G}$ if and only if X is the common zero of the polynomials in P. In [DJK05] they show that the set P can be effectively computed.

The second idea is that checking whether a first-order formula over reals is true, is decidable with Tarski-Seidenberg elimination methods (see for example: [BPR96] and [Ren92]).

Assume that $P = \{p_1, \dots, p_k\} \subset \mathbb{C}[x_{1,1}, \dots, x_{n,n}]$ are the polynomials whose common zeros are exactly the elements of $\bar{G}$. Then, given a quantum automaton $\mathcal{QA}$ and cutpoint

[5]By Euclidean closure we mean that we close the group under taking limits: let $X \in \bar{G}$ then if $\lim_n X^n$ exists it is in $\bar{G}$.

λ, deciding the emptiness of $\mathcal{L}_{>}(\mathcal{QA}, \lambda)$ will be reduced to deciding the truthiness of this statement:

$$\forall X \ , \ (p_1(X) = 0 \wedge p_2(X) = 0 \wedge \cdots \wedge p_k(X) = 0) \implies ||\mathbf{i}XP||^2 - \lambda > 0.$$

Note that $||\mathbf{i}XP||^2$ is a polynomial on the elements of X. Remember that the same can not be used to decide the emptiness of $\mathcal{L}_{\geq}(\mathcal{QA}, \lambda)$ because we took the euclidean closure $\bar{G}$.

The main problem lies in computing the set P of polynomials, the algorithm given in [BJKP05] does not give rise to a complexity bound, while for the algorithm in [DJK05] the authors decide to be more general instead of optimizing the algorithm, and no complexity bound is known. We know that it is at least in *EXPTIME* because computation of Gröbner bases is required and the latter is exponential in the size of the solutions of the system of polynomials. However we conjecture that it can be done in *PSPACE*.

A similar first-order statement can be given for the isolated cutpoint problem.

4.3 Application to probabilistic automata

Since the value 1 problem is decidable for quantum automata, a natural question is: can we use the same techniques to enlarge the leaktight class? We have not pursued this question very far, but we saw that, at least directly, this is not possible.

The first restriction that we need is that the transition matrices should be invertible, in order for them to generate a group. Note that orthogonal matrices are always invertible, and the transition matrices for quantum automata are orthogonal.

The next property that we need is the following: for the group G generated by the transition matrices, we have that if $X \in G$ then, if $\lim_n X^n$ exists it should be invertible. We need this in order for the euclidean closure $\bar{G}$ to be a group. This is what causes the big restriction:

Theorem 4.11. *Let X be the Markov chain induced by the finite word $w \in A^*$ in some given PA $\mathcal{A}$. If $\lim_n X^n$ exists and is invertible then the Markov chain X has no transient states.*

Proof. Assume that it has a transient state t, then t has no incoming edges in $X' = \lim_n X^n$. This means that the column t in the matrix X' is the 0 column, then $det(X') = 0$, a contradiction. □

And these automata for which for every word $w \in A^*$ the induced Markov chain has no transient states is trivially leak tight.

Chapter 5

Conclusion and future work

Simple automata were motivated from the decomposition separation theorem (Theorem 2.4). A surprising result from probability theory about the non-homogeneous Markov chains, that tells us something about the run even though the Markov chain is not homogeneous. This is a natural direction to look when studying PAs on infinite words.

The leaktight property comes as a necessary condition for proving that the Markov monoid of some automaton is complete when looking at the #-factorization tree of the words.

For comparing them we look for a sufficient condition that makes the process induced by some word *not* simple. And after, we look for a witness of such a condition in the Markov monoid, we call this the non-simplicity witness. In the end we prove that if the automaton has a leak then it also has a non-simplicity witness, proving in this way that simple automata are a subset of the leaktight automata.

The future directions of this work can go to enlarging the leaktight automata, perhaps by doing some kind of quantitative analysis. We know that the value 1 problem depends on the transition probabilities, and not only their positiveness from the example in Figure 1.3. Previous work in the literature that we can look upon is some decidable results about *measure once* quantum automata in [DJK05] and [BJKP05]. Surprisingly for this type of automaton the emptiness of the language defined as all the words greater than some cutpoint is undecidable, but for the language of all words *strictly* greater than some cutpoint is decidable.

Also the isolated cutpoint problem for this type of quantum automata is decidable. This is a motivation for searching similar techniques in order to extend the class of leaktight automata. In [DJK05] and [BJKP05] they use tools from algebraic geometry to decide the isolated cutpoint problem. In particular they compute the Zariski closure of the group generated by the transition matrices. That is, they compute a finite set of polynomials of finite degree such that a matrix is an element of this group (meaning that there exists some finite word with this transition matrix) if and only if it is a common zero of the computed set of polynomials, called the Zariski closure.

One can look into using these powerful techniques into probabilistic automata. Using them in a straightforward manner does not seem to be feasible, reason being that the class of PAs becomes too restrictive.

Bibliography

[BF64] David Blackwell and David Freedman. The tail σ–field of a Markov chain and a theorem of Orey. *Ann. Math. Statist.*, 35(3):1291–1295, 1964. Available at http://dx.doi.org/10.1214/aoms/1177703284. MR 0164375. Zbl 0127.35204.

[BJKP05] Vincent D Blondel, Emmanuel Jeandel, Pascal Koiran, and Natacha Portier. Decidable and undecidable problems about quantum automata. *SIAM Journal on Computing*, 34(6):1464–1473, 2005.

[BMT77] Alberto Bertoni, Giancarlo Mauri, and Mauro Torelli. Some recursively unsolvable problems relating to isolated cutpoints in probabilistic automata. In *Automata, languages and programming*, pages 87–94. Springer, 1977.

[BPR96] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM (JACM)*, 43(6):1002–1045, 1996.

[Coh89] Harry Cohn. Products of stochastic matrices and applications. *International Journal of Mathematics and Mathematical Sciences*, 12(2):209–233, 1989.

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

[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.

[CT11] Krishnendu Chatterjee and Mathieu Tracol. Decidable problems for probabilistic automata on infinite words. *CoRR*, abs/1107.2091, 2011.

[DJK05] Harm Derksen, Emmanuel Jeandel, and Pascal Koiran. Quantum automata and algebraic groups. *Journal of Symbolic Computation*, 39(3):357–371, 2005.

[FGO11] Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. A class of probabilistic automata with a decidable value 1 problem. *CoRR*, abs/1104.3055, 2011.

[GO10] Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: decidable and undecidable problems. In *Proceedings of the 37th international colloquium conference on Automata, languages and programming: Part II*, ICALP'10, pages 527–538, Berlin, Heidelberg, 2010. Springer-Verlag.

[KW97] Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pages 66–75. IEEE, 1997.

[MC00] Cristopher Moore and James P Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1):275–306, 2000.

[Ner58] Anil Nerode. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958.

[OVL90] A.L. Oniščik, È.B. Vinberg, and D.A. Leites. *Lie groups and algebraic groups.* Springer Series in Soviet Mathematics Series. Springer London, Limited, 1990.

[Paz71] A. Paz. *Introduction to probabilistic automata.* Computer science and applied mathematics. Academic Press, 1971.

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

[Ren92] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of symbolic computation*, 13(3):255–299, 1992.

[Sim90] Imre Simon. Factorization forests of finite height. *Theor. Comput. Sci.*, 72(1):65–94, 1990.

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

[Son96] Isaac Sonin. The asymptotic behaviour of a general finite nonhomogeneous markov chain (the decomposition-separation theorem). *Lecture Notes-Monograph Series*, pages 337–346, 1996.

[SS61] P. SCHUTZENBERGER and NORTH CAROLINA UNIV CHAPEL HILL DEPT OF STATISTICS. *On the Definition of a Certain Class of Automata.* Defense Technical Information Center, 1961.

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

[Tur69] Paavo Turakainen. Generalized automata and stochastic languages. *Proceedings of the American Mathematical Society*, 21(2):303–309, 1969.

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