Probabilistic automata and Markov chains

Amaury Pouly
Université de Paris, CNRS, IRIF

Lectures notes of the Master Parisien de Recherche en Informatique
Course 2.16 – Finite automata based computation models
Academic year 2020 – 2021
These lectures notes are are intended to be mostly self-contained. As much as possible, I try to use similar notations to Jacques Sakarovitch’s former part of the course on weighted automata and transducers [Sak18].
1 Probabilistic automata

Probabilistic automata are a generalization of finite automata introduced by [Rab63]. They are also a particular case of weighted automata where the weights are rational (or real) and the transition matrices are stochastic (probabilities sums to 1). The interpretation of this model is that the automaton associates to each word a probability of acceptance.

A matrix $M \in \mathbb{R}^{P \times Q}$ is said to be stochastic if all entries are between 0 and 1, and the sum of all entries on each row is equal to 1, i.e. $\sum_{q \in Q} M_{pq} = 1$ for all $p \in P$. A probabilistic automaton is a tuple $A = (A,Q,S,\mu,T)$ where

- $A$ is a finite alphabet,
- $Q$ is a finite set of states,
- $S \in [0,1]^{|A| \times |Q|}$ is stochastic (row) vector of initial probabilities,
- $T \in \{0,1\}^{Q \times 1}$ is a 0–1 (column) vector of accepting states,
- $\mu(a) \in [0,1]^{Q \times Q}$ is a stochastic matrix of transition probabilities, for every $a \in A$.

Unless otherwise stated, we always require the probabilities to be rational numbers. We naturally extend $\mu$ to define a morphism from the set of words to the set of $Q \times Q$ matrices, using the usual matrix product: $\mu(w_1 \cdots w_n) = \mu(w_1) \cdots \mu(w_n)$. To every word $w \in A^*$, we can now associate the probability of acceptance $A(w) = S\mu(w)T$.

It will occasionally be useful to more fin-grained probabilities. Given two states $q,q' \in Q$ and a word $w \in A^*$, we define the probability of going from state $q$ to state $q'$ by reading $w$ to be $A(q \xrightarrow{w} q') = \mu(w)_{q,q'}$.

**Example 1** (Automaton of Figure 1a). We use the notation $a|q$ on an edge of $q$ to $q'$ to signify that the transition labelled by $a$ has probability $p$; formally $\mu(a)_{qq'} = p$. If the probability is 1 then we sometimes write just $a$. In this example, there is a unique initial state, labelled by an incoming arrow, which therefore has probability 1. We identified the accepting states by an extra circle.

This probabilistic automaton is represented by the tuple $A = (A,Q,S,\mu,T)$ where $A = \{a,b\}$, $Q = \{1,2,3\}$ and

$$S = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \mu(a) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \end{bmatrix}, \quad \mu(b) = \begin{bmatrix} \frac{1}{3} & 0 & \frac{2}{3} \\ 1 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{3}{4} \end{bmatrix} \text{ and } T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Consider the word $bb$, it can be accepted through two paths:

- $1 \rightarrow 1 \rightarrow 3$ with probability $\frac{1}{3} \cdot \frac{2}{3} = \frac{2}{9}$
- $1 \rightarrow 3 \rightarrow 3$ with probability $\frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2}$

Note that the path $1 \rightarrow 3 \rightarrow 2$ has probability $\frac{1}{3} \cdot \frac{1}{4}$ but ends at 2 which is not a accepting state. Thus the probability of acceptance of $bb$ is $\frac{2}{9} + \frac{1}{2} = \frac{13}{18}$.

**Exercise 2.** In Example 1, check that $I$, $\mu(a)$ and $\mu(b)$ are stochastic. Check that the acceptance probability of $bb$ matches the formal definition, i.e. $S\mu(bb)T = \frac{13}{18}$. What is the acceptance probability of $aabb$?

**Exercise 3.** Show that the product of two stochastic matrices is stochastic.
Exercise 4. Given \( \langle A, Q, S, \mu, T \rangle \), two states \( q, q' \in Q \) and a word \( w \), what is the interpretation of \( \mu(w)_{q,q'} \)? Prove it. Therefore what is the meaning of \( S\mu(w) \)?

It is often convenient to create probabilistic automata where the transition matrix is not stochastic because the probabilities sum to less than 1. This is the case in Figure 1c: the probability to leave state 2 is only \( \frac{1}{2} \). This can handled in two ways: either by allowing substochastic matrices, where the sum in each row less or equal to 1. Or, by adding a sink state which is not accepting and collects all the missing probabilities. The two approaches are equivalent: any path that reaches the sink state will never leave it and thus has probability of acceptance 0.

Exercise 5. Illustrate the substochastic and sink state approaches on \( C \) from Figure 1c. Show that indeed every word has the same probability in each approach.

In the context of weighted automata, it is natural to consider the weighted language of words recognized by an automaton, where the weight is the probability of acceptance. In the context of probabilistic automata, a new interesting notion of language emerges. Let \( \mathcal{A} \) be a probabilistic automaton and \( 0 \leq \lambda \leq 1 \), define the language recognized by \( \mathcal{A} \) as

\[
\mathcal{L}_\mathcal{A}(\lambda) = \{ w \in A^* : \mathcal{A}(w) > \lambda \}.
\]

In other words, \( \mathcal{L}_\mathcal{A}(\lambda) \) is the set of words accepted by \( \mathcal{A} \) with probability at least \( \lambda \). Any such \( \mathcal{L}_\mathcal{A}(\lambda) \) is called a stochastic language and \( \lambda \) is called a cut-point. Note however that \( \mathcal{L}_\mathcal{A}(\lambda) \) is not a weighted language, there is one language for each \( \lambda \). More generally, given \( \in \{\geq, >, =, 
ot=, <, \leq\} \) a comparison operator, we can consider

\[
\mathcal{L}_\mathcal{A}^\in(\lambda) = \{ w \in A^* : \mathcal{A}(w) \in \in \lambda \}.
\]

Example 6 (Automaton of Figure 1a). We have seen in Example 1 that \( \mathcal{A}(bb) = \frac{13}{18} \) thus \( bb \in \mathcal{L}_\mathcal{A}(\lambda) \) for every \( \lambda < \frac{13}{18} \), but \( bb \notin \mathcal{L}_\mathcal{A}(\lambda) \) for every \( \lambda \geq \frac{13}{18} \).

Exercise 7 (Automaton of Figure 1a). Find a word that is not in \( \mathcal{L}_\mathcal{A}(\frac{1}{2}) \) and one that is in \( \mathcal{L}_\mathcal{A}(\lambda) \) for all \( \lambda < \frac{2}{3} \). Can you find a word in \( \mathcal{L}_\mathcal{A}(\frac{2}{3}) \)? Find an infinite regular language that is included in \( \mathcal{L}_\mathcal{A}(\lambda) \) for all \( \lambda < \frac{2}{3} \).

Exercise 8 (Automata of Figure 1). What is the relationship between \( B \) of Figure 1b and \( A \) of Figure 1a, in particular can you relate \( \mathcal{L}_\mathcal{A}(\lambda) \) and \( \mathcal{L}_B(\lambda) \)?

1.1 Relation to regular languages

It is natural to try to understand how stochastic languages compares to other classes of language, and in particular if they are decidable language. A first simple step toward this goal is to compare them to regular languages.

Exercise 9. Prove that every regular language is stochastic. Hint: take a finite automaton and consider its transition matrix: \( \mu(a)_{q,q'} = 1 \) if there is an edge from \( q \) to \( q' \) labelled by a, 0 otherwise.

Exercise 10. Let \( A \) be a finite alphabet, prove that the collection of regular languages over \( A^* \) is countable.

1.1.1 Non-regular stochastic languages

A first observation is that there exist some stochastic languages that are not regular, this was proven in [Rab63] using a counting argument.

Theorem 11. Stochastic languages strictly contains regular languages.

Proof. Every regular language is stochastic, see Exercise 9. Conversely, we will construct a nonregular stochastic language. Consider \( \mathcal{A} = \langle A, Q, S, \mu, T \rangle \) where \( A = \{0, 1\}, Q = \{p, q\} \) and

\[
S = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad \mu(0) = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad \mu(1) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
\]

This automaton is illustrated in Figure 2a. Given a word \( w \in A^* \), define \( \lambda(w) = \sum_{i=1}^{\lfloor w \rfloor} 2^i \). Indeed, check that \( [\varepsilon] = 0 \), \( [\varepsilon 0] = \frac{w}{2} \) and \( [\varepsilon 1] = \frac{1+w}{2} \). Then we can check by induction that

\[
S = \begin{bmatrix} 1 - w & \varepsilon \\ 1 - \varepsilon \end{bmatrix}, \quad \mu(0) = \begin{bmatrix} 1 - \frac{1}{2}w & \frac{1}{2} \varepsilon \\ 1 - \frac{1}{2}w & \frac{1}{2} \varepsilon \end{bmatrix}, \quad [1 - w] \mu(1) = \begin{bmatrix} \frac{1-w}{2} & \frac{1+w}{2} \end{bmatrix}.
\]

It follows that the probability of acceptance of \( w \) is \( \mathcal{A}(w) = \lfloor w \rfloor \). But now note that \( \lfloor w \rfloor \) is dense in \( [0, 1] \) for \( w \in A^* \). It follows that if \( \lambda < \mu \) then \( \mathcal{L}_\mathcal{A}(\lambda) \supsetneq \mathcal{L}_\mathcal{A}(\mu) \). Indeed, by density we can find \( w \) such that \( \lambda < \mathcal{A}(w) \leq \mu \) since \( \lambda < \mu \). Therefore the collection \( \{ \mathcal{L}_\mathcal{A}(\lambda) : \lambda \in [0, 1] \} \) is uncountable. But the collection of regular languages is countable, thus there exists a \( \lambda \) such that \( \mathcal{L}_\mathcal{A}(\lambda) \) is not regular.

\[\square\]
We will need the following classical theorem on the characterization of regular languages.

**Theorem 12** (Myhill–Nerode). Let \( L \) be a language, we say that two words \( u \) and \( v \) are right equivalent for \( L \), and write \( u \equiv_L v \), if for every \( w \in A^* \), we have \( uw \in L \) if and only if \( vw \in L \). Prove that \( \equiv_L \) is an equivalence relation. Show that a language \( L \) is regular if and only if the number of equivalence classes of \( A^* \) with respect to \( \equiv_L \) is finite. Furthermore, the number of equivalence classes corresponds to the number of states of the smallest deterministic finite automaton that recognizes \( L \).

**Exercise 13.** Prove Theorem 12. For the last statement, you can show that the number of equivalence classes is a bound of the number of states (and not necessarily that it is optimal). *Hint:* the equivalence classes correspond to states of an automaton that recognizes \( L \).

While the above theorem is very much non-constructive, much more is actually known now about the automaton defined the proof:

**Theorem 14.** There exists a probabilistic automaton \( A \) such that for every \( \lambda \in [0, 1] \), \( \mathcal{L}_A(\lambda) \) is regular if and only if \( \lambda \) is rational.

**Proof.** Let \( A \) be the automaton from the proof of Theorem 11. Assume that \( \lambda \) is rational, then \( \lambda \) has an eventually periodic binary expansion (note that if \( \lambda \) has a finite binary expansion, we can take \( v = 0 \) to make it infinite). Therefore we have \( \lambda = [uvwv \cdots] = [uv^\omega] \) for some \( u, v \in \{0, 1\}^* \). It is then immediate to see that \( \mathcal{L}_A(\lambda) \) is regular. Indeed, if we read a word \( w \in \{0, 1\}^* \),

\[ [w] = \mathcal{L}_A(\lambda). \]

\( \square \)

### 1.1.2 Universally non-regular probabilistic automata

The original construction by Rabin showed that there exists an automaton \( A \) such that \( \mathcal{L}_A(\lambda) \) is not regular for at least one \( \lambda \). Surprisingly, a small modification of this automaton given by [FS15] allows us to strengthen this statement.

**Theorem 15.** There exists a universally non-regular probabilistic automaton, i.e. an automaton \( B \) such that \( \mathcal{L}_B(\lambda) \) is non-regular for all \( \lambda \in (0, 1) \).

**Proof.** Consider automaton \( B \) illustrated in Figure 2b, it is defined over the alphabet \( A' = A \cup \{\bar{z}\} = \{0, 1, \bar{z}\} \). It is the same as automaton \( A \) from the proof of Theorem 11 with an extra transition from \( q \) to \( p \) labelled by \( \bar{z} \). Note that when reading a \( \bar{z} \), the automaton must be in state \( q \) for the word to be accepted with positive probability. Therefore for all \( u, v \in A^* \), we have that

\[ \mathcal{B}(uv) = B \left( p \xrightarrow{u} q \right) B \left( p \xrightarrow{v} q \right) = \mathcal{A}(u)\mathcal{A}(v) = [uv]. \]

Recall that the set \( [A^*] = \{[w] : w \in A^*\} \) is dense in \([0, 1]\).

Now fix \( \lambda \in (0, 1) \) and take \( u, v \in A^* \) such that \( \lambda < |w| < |v| \). Then by density of \( [A^*] \) we can find \( w \in A^* \) such that \( \lambda / |w| > |w| > \lambda / |v| \). But then \( \mathcal{B}(uv) = [uv] \) whereas \( \mathcal{B}(uv) = [uv] > \lambda \). This shows that \( u \not\equiv_{\mathcal{L}_B(\lambda)} v \). Again by density of \( [A^*] \), we can find infinitely many such pairs \( u, v \) and thus \( \mathcal{L}_B(\lambda) \) cannot be regular by Theorem 12. \( \square \)

**Exercise 16.** Let \( C = \langle A, Q, S, \mu, T \rangle \) be the automaton illustrated in Figure 2c. Give \( A, Q, S, \mu \) and \( T \). Show for every word \( x(n_1, \ldots, n_k) = a_{n_1}ba_{n_2}\cdots a_{nk} \) we have \( C(x(n_1, \ldots, n_k)) = 2^{-m} \prod_{i=1}^{k} (1 - 2^{-n_i}) \). Show that if \( u = x(n_1, \ldots, n_k) \) and \( w = x(n_{k+1}, \ldots, n_k) \) then \( C(uw) = C(u)C(w) \). Show that \( \{C(x(n_1, \ldots, n_k)) : n_1, \ldots, n_k \in \mathbb{N}, k \in \mathbb{N} \} \) is dense in \([0, 1]\).

Conclude that \( C \) is universally non-regular. *Hint:* use the same proof idea as Theorem 15.
1.1.3 Isolated cut-points

An interesting observation in the examples above is that stochastic languages that are non-regular tend to verify that $L_A(\lambda) \neq L_A(\lambda + \epsilon)$ for small $\epsilon$. For example, this was essential in the proof of existence of such languages. On the other hand, simple examples that only recognize regular language tend to satisfy the opposite property that the language is unchanged by small perturbation in the threshold. The latter are called isolated cut-points.

Formally, a cut-point $\lambda$ is called isolated with respect to some probabilistic automaton $A$ if there exists $\delta > 0$ such that

$$|A(w) - \lambda| \geq \delta, \forall w \in A^*.$$  

We will call $\delta$ the isolation threshold (for $\lambda$), although there is no standard name for it.

Theorem 17. If $\lambda$ is isolated with respect to $A$ then $L_A(\lambda)$ is regular. Furthermore, if $A$ has $n$ states and $r$ final states, then $L_A(\lambda)$ can be recognized by a finite deterministic automaton with at most $(1 + \frac{\epsilon}{\delta})^{n-1}$ states where $\delta$ is the isolation threshold.

Proof. First assume there is a unique final state. Write $A = (A, Q, S, \mu, T)$ where $Q = \{s_1, \ldots, s_n\}$ and $s_n$ is the only final state. Let $L = L_A(\lambda)$ and assume that $\lambda$ is isolated with threshold $\delta > 0$. Let $x_1, \ldots, x_k \in A^*$ be pairwise $\equiv_L$ inequality words (see Theorem 12). Then by definition, for every $i \neq j$, there exist $y \in A^*$ such that $x_i y \in L$ but $x_j y \notin L$ (or the other way around). Since $\lambda$ is isolated we must have that

$$A(x_i y) - A(x_j y) \geq 2\delta.$$  

Let $(\xi_1, \ldots, \xi_n)$ be the first row of $\mu(x_i)$. Let $(\eta_1, \ldots, \eta_n)$ be the last column of $\mu(y)$, for this particular $y$. Check that $A(x_i y) = S \mu(x_i y) T = S \mu(x_i) \mu(y) T = \xi_1 \eta_1 + \cdots + \xi_n \eta_n$ and thus

$$A(x_i y) - A(x_j y) = (\xi_i - \xi_j) \eta_1 + \cdots + (\xi_i - \xi_j) \eta_n \geq 2\delta.$$  

But since $\mu(y)$ is stochastic, we must have $0 \leq \eta_\ell \leq 1$ for all $\ell$. This implies that

$$|\xi_i - \xi_j| + \cdots + |\xi_i - \xi_j| \geq 2\delta, \quad \text{for } i \neq j.$$  

In other words, the points $\xi_i$ and $\xi_j$ cannot be too close to each other for the $L^1$ norm. Coupled with the fact that they are stochastic vectors (and thus live in $[0,1]^n$), this will put a bound on $k$.

Let $||x|| = |x_1| + \cdots + |x_n|$ denote the $L^1$ norm and $B_R(p) = \{x \in \mathbb{R}^n : ||x - p|| < R\}$ denote the $L^1$ open ball of radius $R$ and center $p \in \mathbb{R}^n$. We will use the fact that $B_R(p)$ has volume $c R^n$ where $c$ only depends on $n$ (in fact $c = \frac{\pi^n}{(n-1)!}$). Now we can rephrase (1) as $||\xi_i - \xi_j|| \geq 2\delta$ which implies that $B_\delta(\xi_i) \cap B_\delta(\xi_j) = \emptyset$ for $i \neq j$. On the other hand, by stochasticity, we have that $||\xi_i|| = 1$ thus if $x \in B_\delta(\xi_i)$ then $||x|| \leq ||x - \xi_i|| + ||\xi_i|| < 1 + \delta$ thus $B_\delta(\xi_i) \subseteq B_{1+\delta}(0)$. Therefore,

$$B_\delta(\xi_1) \cup \cdots \cup B_\delta(\xi_k) \subseteq B_{1+\delta}(0)$$

where $\cup$ denotes the disjoint union. By taking the volume, we get that $k c \delta^n \leq c (1+\delta)^n$ and thus

$$k \leq (1 + \frac{\delta}{\pi})^n.$$  

This show that the number of equivalence classes with respect to $\equiv_L$ is finite and therefore $L$ is regular. Furthermore this gives us a bound on the number of states by Theorem 12. It is possible to improve the bound further by noting that the $\xi_i$ are stochastic vectors, therefore they belong to the hyperplane $H$ defined by $x_1 + \cdots x_n = 1$, which is a $n-1$ dimensional subspace. Therefore we get that

$$(B_\delta(\xi_1) \cap H) \cup \cdots \cup (B_\delta(\xi_k) \cap H) \subseteq B_{1+\delta}(0) \cap H$$

where all intersections with $H$ are nonempty. We conclude by noticing that if $B_R(p) \cap H$ is nonempty, its volume in $H$ is $c' R^{n-1}$ (where in fact $c' = \frac{\sqrt{\pi}}{\sqrt{2\pi} n}$).  

The previous theorem suggests that finding an equivalent deterministic finite automaton might increase the number of states. Furthermore, the increase fundamentally depends on the isolation threshold, which we do not know if it is lower bounded by a function of $n$. The next theorem shows that, in fact, it is not.

Theorem 18. There exists a probabilistic automaton $A$ with only two states and a sequence $(\lambda_n)_{n \in \mathbb{N}}$ of isolated cut points such that $L_A(\lambda_n)$ cannot be recognized by a deterministic finite automaton with less than $n$ states.
Proof. Let \( A = \langle A, Q, S, \mu, T \rangle \) where \( A = \{0, 2\} \), \( Q = \{s_0, s_1\} \) and

\[
S = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mu(0) = \begin{bmatrix} \frac{1}{3} & 0 \\ \frac{1}{3} \end{bmatrix}, \quad \mu(2) = \begin{bmatrix} \frac{2}{3} & \frac{2}{3} \\ 1 \end{bmatrix}, \quad \text{and} \quad T = \begin{bmatrix} 0 & 1 \end{bmatrix}.
\]

This automaton is very similar to the dyadic automaton of Figure 2a but in base 3. Similarly, we get that if \( w \in A^* \) then

\[
A(w) = \sum_{i=1}^{|w|} w_i 3^{|w|-i-1}.
\]

Contrary to the previous automaton, the set \( P = \{ A(w) : w \in A^* \} \) is not dense anymore. Indeed, \( P \) is included in the Cantor set \( C \), that is exactly the set of real numbers of \([0, 1]\) that do not require the digit 1 in their ternary (base 3) expansion. In fact, it can be seen that \( C \) is the topological closure of \( P \), therefore if \( \lambda \notin C \) then \( \lambda \) must be an isolated cut-point of \( A \). Now fix \( n \in \mathbb{N} \) and consider the cut point

\[
\lambda_n = 0.2222\cdots 2211 = \sum_{i=1}^{n-1} 2 \cdot 3^{-i} + 3^{-n} + 3^{-n-1}.
\]

Note that \( \lambda_n \notin C \) because its ternary expansions\(^1\) contain the digit 1. Then the word \( 2^n \in L_A(\lambda_n) \) since it has probability of acceptance

\[
A(2^n) = \sum_{i=1}^n 2 \cdot 3^{-i-1} > \lambda_n.
\]

Conversely, if \( w \in A^* \) has length \(|w| \leq n - 1\) then

\[
A(w) \leq \sum_{i=1}^{|w|} 2 \cdot 3^{-i-1} \leq \sum_{i=1}^{n-1} 2 \cdot 3^{-i} \leq \lambda_n.
\]

It follows that \( A(\lambda_n) \) is nonempty and must reject all words of length less than \( n \). Therefore any deterministic finite automaton that recognizes this language must have at least \( n \) states (see Exercise 19). \( \square \)

**Exercise 19.** Let \( L \) be a nonempty regular language that contains no words of length less than \( n \). Show that any deterministic finite automaton that recognizes \( L \) must have at least \( n \) states. *Hint: use Theorem 12.*

### 1.2 Operations on probabilistic automata

Probabilistic automata naturally define functions from \( \Sigma^* \) to \([0, 1]\). This gives us more structure to work with than classical automata and begs the question of which operations can be done effectively. Natural operations include:

- convex combinations: \( \alpha A(w) + (1 - \alpha) B(w) \);
- complement: \( 1 - A(w) \);
- product: \( A(w) B(w) \);
- changing the probability of the empty word.

We will now see that all these operations are effective.

**Lemma 20.** For any two probabilistic automata \( A \) and \( B \) over the same alphabet, and \( \alpha \in [0, 1] \), there exists an automaton \( \alpha A + (1 - \alpha) B \) that satisfies \((\alpha A + (1 - \alpha) B)(w) = \alpha A(w) + (1 - \alpha) B(w)\) for every word \( w \).

**Proof.** Let \( A = \langle A, Q, S, \mu, T \rangle \) and \( B = \langle A, Q', S', \mu', T' \rangle \). Define \( C = \langle A, Q \cup Q', S'', \mu'', T'' \rangle \) where

\[
S'' = \left[ \begin{array}{c} \alpha S \ (1 - \alpha) S' \end{array} \right], \quad \mu''(a) = \left[ \begin{array}{c} \mu(a) \ 0 \ \mu'(a) \end{array} \right], \quad T'' = \left[ \begin{array}{c} T \\ T' \end{array} \right].
\]

One can then check that

\[
C(w) = \left[ \begin{array}{c} \alpha S \ (1 - \alpha) S' \end{array} \right] \left[ \begin{array}{c} \mu(w) \ 0 \ \mu'(w') \end{array} \right] \left[ \begin{array}{c} T \\ T' \end{array} \right] = \alpha S \mu(w) T + (1 - \alpha) S' \mu'(w) T'.
\]

Graphically, this construction corresponds to the following:

\(^1\)Beware that a number can have two ternary expansions, for example \( \frac{1}{3} = 0.1 = 0.0222\cdots \) in base 3. In this case, \( \lambda_n = 0.2222\cdots 2211 = 0.2222\cdots 220222\cdots \) that both use the digit 1.
Lemma 21. For any probabilistic automaton $A$, there exists $A^c$ such that $A^c(w) = 1 - A(w)$ for every word $w$.

Proof. Let $A = \langle A, Q, S, \mu, T \rangle$ be a probabilistic automaton and define $A^c = \langle A, Q, S, \mu, T' \rangle$ where $T'_i = 1 - T_i$, i.e. $S$ is the “complement” over $T$. Then for every word $w \in A^*$, we have that

$$A^c(w) = S\mu(w)S = \sum_{i=1}^{\lvert Q \rvert} (S\mu(w))_i T_i = \sum_{i=1}^{\lvert Q \rvert} (S\mu(w))_i (1 - T_i) = \sum_{i=1}^{\lvert Q \rvert} (S\mu(w))_i - \sum_{i=1}^{\lvert Q \rvert} (S\mu(w))_i T_i = 1 - A(w)$$

by stochasticity of $S\mu(w)$.

Lemma 22. For any two probabilistic automata $A$ and $B$ over the same alphabet, there exists a product automaton $A \cdot B$ that satisfies $A \cdot B(w) = A(w)B(w)$ for every word $w$.

Proof. Let $A = \langle A, Q, S, \mu, T \rangle$ and $B = \langle A, Q', S', \mu', T' \rangle$. Define $C = \langle A, Q \times Q', S \otimes S', \mu \otimes \mu', T \otimes T' \rangle$ where $\otimes$ is the Kronecker product: given $M \in \mathbb{R}^{I \times J}$ and $M' \in \mathbb{R}^{I' \times J'}$ then $M \otimes M' \in \mathbb{R}^{(I \times I') \times (J \times J')}$ is defined by

$$(M \otimes M')(i,i'),(j,j') = M_{i,j}M'_{i',j'}.$$ 

We check that $\otimes$ preserves stochasticity: for every $(i, i') \in I \times I'$,

$$\sum_{j \in J} \sum_{j' \in J'} (M \otimes M')(i,i'),(j,j') = \sum_{j \in J} M_{i,j} \sum_{j' \in J'} M'_{i',j'} = \sum_{j \in Q} M_{i,j} = 1$$

if $M$ and $M'$ are stochastic. And furthermore, it satisfies the mixed-product property:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

Indeed, if $A \in \mathbb{R}^{I \times K}, B \in \mathbb{R}^{I' \times K'}, C \in \mathbb{R}^{K \times J}, D \in \mathbb{R}^{K' \times J'},$

$$(A \otimes B)(C \otimes D)_{(i,i'),(j,j')} = \sum_{k \in K} \sum_{k' \in K'} (A \otimes B)(i,i'),(k,k')(C \otimes D)(k,k'),(j,j') = \sum_{k \in K} \sum_{k' \in K'} A_{i,k}B_{i',k'}C_{k,j}D_{k',j'} = \sum_{k \in K} A_{i,k}C_{k,j} \sum_{k' \in K'} B_{i',k'}D_{k',j'} = (AC)_{i,j}(BD)_{i',j'} = ((AC) \otimes (BD))_{(i,i'),(j,j')}.$$ 

Therefore for every word $w \in A^*$ we have that

$$C(w) = (S \otimes S')(\mu''(a)(T \otimes T')) = (S \otimes S'((\mu(a_1) \otimes \mu'(a_1)) \cdots (\mu(a_n) \otimes \mu'(a_n)))(T \otimes T') = (S\mu(a_1) \cdots \mu(a_n)T) \otimes (S\mu(a_1) \cdots \mu(a_n)T)$$

by the mixed-product property

$$= A(w)B(w).$$

In a number of proofs, we will want to specifically change the probability of the empty word after a construction, typically to change it to zero so that it is rejected.

Lemma 23. For any probabilistic automaton $A$ and probability $p$, there exists an automaton $A[\varepsilon \leftarrow p]$ that satisfies $A[\varepsilon \leftarrow p](\varepsilon) = p$ and $A[\varepsilon \leftarrow p](w) = A(w)$ for any non-empty word $w$. 


Proof. Let $A = \langle A, Q, S, \mu, T \rangle$ and let $q_e, \tilde{q}_e \notin Q$ be a fresh states. Define $B = \langle A, \{q_e, \tilde{q}_e\} \cup Q, S', \mu', T' \rangle$ where

$$S' = \begin{bmatrix} p & 1 - p & 0 \end{bmatrix}, \quad \mu'(a) = \begin{bmatrix} 0 & 0 & S\mu(a) \\ 0 & 0 & S\mu(a) \\ 0 & 0 & \mu(a) \end{bmatrix}, \quad T' = \begin{bmatrix} 1 \\ 0 \\ T \end{bmatrix}.$$  

Graphically, this construction corresponds to the following:

It is clear that $\mu'(a)$ is stochastic and furthermore we have that

$$B(\varepsilon) = S'T' = \begin{bmatrix} p & 1 - p & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ T \end{bmatrix} = p.$$  

Furthermore, check by induction that for every non-empty word $w$, we have

$$\mu'(w) = \begin{bmatrix} 0 & 0 & S\mu(w) \\ 0 & 0 & S\mu(w) \\ 0 & 0 & \mu(aw) \end{bmatrix}$$  

and hence

$$B(w) = S'(w)T' = \begin{bmatrix} p & 1 - p & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & S\mu(w) \\ 0 & 0 & S\mu(w) \\ 0 & 0 & \mu(aw) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ T \end{bmatrix} = pS\mu(w) + (1 - p)S\mu(w)T = A(w).$$

1.3 The emptiness problem and its variants

Given a stochastic language, the first question that comes to mind is whether this language is empty or not. Surprisingly, even this simple problem turns out to be undecidable. Note that Rabin defines the language $L_1$ as words with probability of acceptance strictly greater than $\lambda$, but some authors prefer to use another convention where the probability is greater or equal to $\lambda$. To avoid any ambiguity, we distinguish the two problems and follow a recent proof strategy [GO10].

Problem 24 (Strict Emptiness). Given a probabilistic automaton $A$ and a cut-point $\lambda$, decide whether there exists a word $w$ such that $A(w) > \lambda$.

Problem 25 (Emptiness). Given a probabilistic automaton $A$ and a cut-point $\lambda$, decide whether there exists a word $w$ such that $A(w) \geq \lambda$.

Problem 26 (Universality). Given a probabilistic automaton $A$ and a cut-point $\lambda$, decide whether it is true that $A(w) \geq \lambda$ for all word $w$.

In fact, all three problems are essentially equivalent and reduce to the following variant of the problem where we look for words with a specific probability of acceptance.

Problem 27 (Equality). Given a probabilistic automaton $A$ and a cut-point $\lambda$, decide whether there exists a word $w$ such that $A(w) = \lambda$.

It is clear that those problems are equivalent to asking whether $L^>\lambda(A)$, $L^\lambda(A)$ and $L^\lambda(A)$ are empty. We will start with the equality problem and reduction from the Post Correspondence Problem (PCP) given by Bertoni. Recall that PCP is a classical example of undecidable problem.

Problem 28 (PCP). Given $A$ a finite alphabet and $\phi_1, \phi_2 : A \to \{0, 1\}^*$ two functions that we naturally extend to morphisms over $A^*$, decide whether there exists $w \in A^* \setminus \{\varepsilon\}$ such that $\phi_1(w) = \phi_2(w)$.

It will be important, for a later reduction, to consider a particular sub-class of probabilistic automaton where only certain probabilities appear. An automaton is called simple if every initial and transition probability is in $\{0, \frac{1}{2}, 1\}$.
Theorem 29. The Equality Problem is undecidable, even for simple automata and cut-point $\frac{1}{2}$.

Proof. We will reduce from the PCP: let $\phi_1, \phi_2 : A \rightarrow \{0,1\}^*$ be an instance. We modify this instance into $\phi_1, \phi_2$ by inserting 1 after every letter of $\phi_i(a)$ so that $\phi_i(a) \in \{0,1,1\}^*$. Clearly, $\phi_1(w) = \phi_2(w)$ if and only if $\phi_1(w) = \phi_2(w)$ so this modification preserves the undecidability.

We will build a probabilistic automaton $A$ such that $A$ accepts a word with probability $\frac{1}{2}$ if and only if this PCP instance has a solution. We do so by encoding $\{0,1\}^*$ into probabilities. Similarly to the proof of Theorem 11, define

$$[w] = \sum_{i=1}^{[w]} w 2^{-i} \quad \text{for every } w \in \{0,1\}^*.$$  

Check that $[\cdot]$ is injective over $\{0,1,1\}^*$ and therefore for every words $w \in A^*$,

$$[\phi_1(w)] = [\phi_2(w)] \text{ if and only if } \phi_1(w) = \phi_2(w).$$  

We can now consider the following two automata for $i \in \{1,2\}$: $A_i = (A, Q, S, \mu, T)$ where $Q = \{p,q\}$ and

$$S = \binom{1}{0}, \quad \mu(a) = \begin{bmatrix} 2^{-[\phi_1(w)]} & [\phi_1(w)] \\ 0 & 1 \end{bmatrix}, \quad \text{and } T = \binom{0}{1}.$$  

One checks that $\mu(a)$ is substochastic by checking that $[w] \leq 1 - 2^{-[w]}$ for every $w \in \{1,0\}^*$. For every $u, v \in \{0,1\}^*$, check that $[uv] = [u] + 2^{-[w]} [v]$. Then check that if $a, b \in A$ we have that

$$\mu_i(a) \mu_i(b) = \begin{bmatrix} 2^{-[\phi_i(a)]} \cdot [\phi_i(b)] & [\phi_i(a)] + 2^{-[\phi_i(a)]} [\phi_i(b)] \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2^{-[\phi_i(ab)]} & [\phi_i(ab)] \\ 0 & 1 \end{bmatrix}$$  

and therefore for every word $w \in A^*$,

$$A_i(w) = S \begin{bmatrix} 2^{-[\phi_i(w)]} & [\phi_i(w)] \\ 0 & 1 \end{bmatrix} T = [\phi_i(w)].$$  

Finally, we can build automaton $B = \frac{1}{2} A_1 + \frac{1}{2} A_2$. We then obtain that every word $w \in A^*$,

$$B(w) = \frac{1}{2} \iff \frac{1}{2} A_1(w) + \frac{1}{2} A_2(w) = \frac{1}{2} \iff [\phi_1(w)] = [\phi_2(w)] \iff \phi_1(w) = \phi_2(w)$$  

using (2). It remains to deal with the empty word since $B(\varepsilon) = \frac{1}{2}$ but we do not want to accept the empty word. Therefore $B[\varepsilon \leftarrow 0]$ accepts a word with probability $\frac{1}{2}$ if and only if the PCP instance has a solution.

In this proof, note that automata $B$ only uses dyadic transition probabilities, and therefore we can make it simple by introducing more states (see Exercise 30). Furthermore, check that the construction of $B[\varepsilon \leftarrow 0]$ preserves simplicity. \hfill $\Box$

Exercise 30. Recall that a dyadic number (or dyadic rational) is a number of the form $a2^{-p}$ for some $a \in \mathbb{Z}$ and $p \in \mathbb{N}$. Given a probabilistic automaton $A$ over alphabet $\{0,1\}$ where all transition probabilities are dyadic, build a simple automaton $B$, also over alphabet $\{0,1\}$, such that $\{B(w) : w \in A^*\} = \{A(w) : w \in A^*\}$. Hint: if $p$ is the maximum dyadic exponent that appears in $A$, build $B$ such that $A(w_1 \cdots w_k) = B(w_1^{p+1} \cdots w_k^{p+1})$.

Proposition 31. Given a simple probabilistic automaton $A$, one can compute (simple) probabilistic automata $B$ and $C$ such that the following propositions are equivalent:

- there exists a word $w$ such that $A(w) = \frac{1}{2}$,
- there exists a word $w$ such that $B(w) \geq \frac{1}{2}$,
- there exists a word $w$ such that $C(w) \geq \frac{1}{2}$.

Proof. The idea of the proof is that $x = \frac{1}{2}$ if and only if $x(1-x) \geq \frac{1}{2}$. Therefore from $A$, build $B = A \cdot A^c$ (see Lemma 22), then $B(w) = A(w)(1-A(w))$. By construction, all transition probabilities of $B$ are already multiple of $\frac{1}{4}$ (see Exercise 32).

To build $C$, we start by noticing that since all initial and transition probabilities of $B$ are multiple of $\frac{1}{4}$, $B(w)$ is a multiple of $4^{-[w]-1}$ for every word $w$. Thus $B(w) \geq \frac{1}{4}$ if and only if $B(w) > \frac{1}{4} - 4^{-[w]-1}$ if and only if $\frac{1}{2} B(w) + \frac{1}{2} 4^{-[w]-1} > \frac{1}{2}$. One easily builds an automaton $D$ such that $D(w) = 4^{-[w]-1}$ for any word $w$. Then $C = \frac{1}{2} B + \frac{1}{2} D$ satisfies that $C(w) > \frac{1}{8} \iff B(w) > \frac{1}{4} - 4^{-[w]-1} \iff B(w) \geq \frac{1}{2}$.
Exercise 32. Show that if $A$ and $B$ are simple then all transition probabilities of $A \cdot B$ are multiple of $\frac{1}{2}$.

As a consequence of Theorem 29 and Proposition 31, we get:

**Theorem 33.** The emptiness and strict emptiness problems are undecidable, even for simple automata and cut-point $\frac{1}{2}$.

**Exercise 34.** Given $A$ a probabilistic automaton and a rational cut-point $\lambda \in (0, 1)$, build automata $B$ and $C$ such that $A(w) \geq \lambda$ (resp. $A(w) > \lambda$) if and only if $B(w) \geq \frac{1}{2}$ (resp. $C(w) \geq \frac{1}{2}$). Show that if $A$ is simple and $\lambda$ is dyadic then you can make $B$ and $C$ simple.

1.4 The isolation problem

We saw in Section 1.1.3 that isolated cut-points are very special since they define regular languages. On the other hand, the emptiness language is undecidable in general for probabilistic automata but decidable for finite automata. Therefore, if we can detect that a cut-point is isolated, it would give us a way to decide emptiness in certain cases.

**Problem 35 (Isolation).** Given a probabilistic automaton $A$ and a cut-point $\lambda$, decide whether $\lambda$ is isolated with respect to $A$.

Unfortunately, this problem is undecidable in general and even when the threshold is fixed. An elegant way to prove this is to reduce to a variant of the PCP problem for infinite words, which is also undecidable.

**Remark 36.** We will see an alternative proof of undecidability in Section 1.6.

**Problem 37 ($\omega$–PCP).** Given $A$ a finite alphabet and $\phi_1, \phi_2 : A \rightarrow \{0, 1\}^*$ two functions that we naturally extend to morphisms over $A^*$, decide whether there exists $w \in A^*$ such that $\phi_1(w) = \phi_2(w)$.

**Exercise 38.** Show that $\omega$–PCP is undecidable.

In particular, we will use a classical feature of the $\omega$–PCP problem: if an instance is not solvable, then there is uniform bound on the how far the first different letter can be.

**Lemma 39.** Let $\phi_1, \phi_2 : A \rightarrow \{0, 1\}^*$ be an instance of the $\omega$–PCP that has no solution. Then there exists $n_0 \in \mathbb{N}$ such that for every infinite (or non-empty finite) word $w$, there exists $i \leq n_0$ such that $\phi_1(w)_i \neq \phi_2(w)_i$. In other words, $\phi_1(w)$ and $\phi_2(w)$ differ already in their first $n_0$ letters, and $n_0$ is independent of $w$.

**Proof.** Consider the tree where the root is labelled $(\varepsilon, \varepsilon)$ and given a node $(u, v)$ of the tree, if $u_i = v_i$ for all $i \leq \min(|u|, |v|)$, then this node has children $(u\phi_1(a), \phi_2(a))$ for all $a \in A$. In other words, we write on the nodes the result of finite labelling of the $\omega$–PCP and we continue only if we haven’t found a differing letter (but labels are allowed to differ in length, in which case we only compare up to the shortest one). This tree is finitely branching since each node has 0 or $|A|$ children and $A$ is finite. This tree has no infinite path for it would imply that this instance of $\omega$–PCP has a solution. Therefore by König’s lemma, the tree is finite. Let $n_0$ be the longest length of a word that appears in a label of the tree. Since the labels of the nodes are of the form $(\phi_1(w), \phi_2(w))$, this shows the result. \qed

**Theorem 40.** The isolation problem is undecidable, even for simple automata and a fixed rational cut-point $0 < \lambda < 1$.

**Proof.** We will show the result for $\lambda = \frac{1}{2}$, this can be extended to any rational $\lambda \in (0, 1)$ by Exercise 34.

The problem will essentially be the same as for Theorem 29 with a twist. Let $\phi_1, \phi_2 : A \rightarrow \{0, 1\}^*$ be an instance of the $\omega$–PCP. We modify this instance so that $\phi_i(w) \in \{0, 1\}^+$ for every non-empty word $w$. This can be done by adding a “1” after each letter of $\phi_i(a)$ for every $a \in A$. Clearly, this does not change the undecidability of $\omega$–PCP.

Like in the proof of Theorem 29, we define $|w| = \sum_{i=1}^{|w|} w_i 2^{-i}$ for every $w \in \{0, 1\}^+$ and build a probabilistic automaton $C$ such that $C(w) = \frac{1}{2} + \frac{1}{2}([\phi_1(w)] - [\phi_2(w)])$ for every $w \in A^*$. Finally we let $B = C[\varepsilon \leftarrow 0]$ to avoid any problem with the empty word. Recall that $|wx| = |w| + 2^{-|w|}|x|$ for all words $w, x$. We will now show that $\frac{1}{2}$ is isolated if and only if this instance of $\omega$–PCP is not solvable.

Assume that this instance has a solution $w \in A^N$. Let $n \in \mathbb{N}$, then there exists a finite prefix $u$ of $w$ such that $|\phi_1(u)| \geq n$ and $|\phi_2(u)| \geq n$ (since $\phi_1(w)$ and $\phi_2(w)$ are infinite words). Since the instance is solvable, $\phi_1(w) = \phi_2(w)$ and thus the first $n$ letters of $\phi_1(u)$ and $\phi_2(u)$ are the same, i.e. $\phi_1(u) = px$ and $\phi_2(u) = py$ for some $p \in A^n$ and $x, y \in A^*$. But then

$$||\phi_1(u) - \phi_2(u)|| = ||px - py||$$

$$= ||p| + 2^{-|p|}|x| - |p| - 2^{-|p|}y||$$

$$= 2^{-n} |x| - |y|$$

$$\leq 2^{1-n}$$

since $|x|, |y| \in [0, 1]$. 

\[\text{Version 0.991}\]
Therefore, since \( w \) is non-empty,
\[
|\mathcal{B}(w) - \frac{1}{2}| = |\mathcal{C}(w) - \frac{1}{2}| = \frac{1}{2} \left| \left| \phi_1(w) \right| - \left| \phi_2(w) \right| \right| \leq 2^{-n}.
\]
This shows that \( \frac{1}{2} \) is not isolated, since there are words accepted with probabilities arbitrarily close to the cut-point.

Conversely, assume that this instance has no solution. Then by Lemma 39, there exists \( n_0 \in \mathbb{N} \) such that for every infinite (or non-empty finite) word \( w \in A^\mathbb{N} \), there exists \( i \leq n_0 \) such that \( \phi_1(w)_i \neq \phi_2(w)_i \). Recall that we modified the instances so that \( \phi_i(w) \in \{0,1\}^\ast 1 \) for every word \( w \). Let \( w \in A^\ast \), then we can write \( \phi_1(w) = u01x \) and \( \phi_2(w) = ub1y \) where \(|u| \leq n_0 \), \( a, b \in \{0, 1\} \) are distincts and \( x, y \in \{0, 1\}^\ast \). It follows that
\[
\left| \left| \phi_1(w) \right| - \left| \phi_2(w) \right| \right| = \left| \left| [u01x] - [ub1y] \right| \right| \\
\geq |a - b| 2^{-|u|} + |[x] - [y]| 2^{-|u| - 2} \\
\geq 2^{-|u|} - 2^{-|u| - 2} \\
\geq 2^{-|u|} \text{ since } |[x], [y] | \in [0, 1] \\
\geq 2^{1-n_0} \text{ since } |u| \leq n_0.
\]

It follows that for all non-empty word \( w \),
\[
|\mathcal{B}(w) - \frac{1}{2}| = |\mathcal{C}(w) - \frac{1}{2}| = \frac{1}{2} \left| \left| \phi_1(w) \right| - \left| \phi_2(w) \right| \right| \geq 2^{-n_0}.
\]
and the empty word has probability 0 so \( |\mathcal{B}(\varepsilon) - \frac{1}{2}| = \frac{1}{2} \geq 2^{-n_0} \). Since \( n_0 \) is independent of \( w \), this shows that \( \frac{1}{2} \) is isolated.

\[\square\]

### 1.5 The value 1 problem

There is a slight discrepancy in Theorem 40 for the case \( \lambda = 0 \) and \( \lambda = 1 \). It is clear that those two cases are symmetric, by taking the complement of the automaton. If we fix the cut-point to 1, the isolation problem is the same asking if there are words accepted with probabilities arbitrarily close to 1. This is related to asking what is the value of an automaton. The value of probabilistic automaton \( \mathcal{A} \) over alphabet \( A \) is
\[
\text{val}(\mathcal{A}) = \sup \{\mathcal{A}(w) : w \in A^\ast\}.
\]
In other words, it is the supremum of the probability of acceptance over all possible input words. Note that this probability may not be achieved by any path, only as the limit of longer and longer paths.

**Problem 41 (Value 1).** Given a probabilistic automaton \( \mathcal{A} \), decide whether \( \mathcal{A} \) has value 1, i.e. for every \( \varepsilon > 0 \), there exists \( w \) such that \( \mathcal{A}(w) > 1 - \varepsilon \).

**Remark 42.** We will see an alternative proof of undecidability in Section 1.6.

\begin{center}
\begin{tikzpicture}
\tikzstyle{node}=[minimum size=1cm,draw,fill=white]
\tikzstyle{edge}=[double]
\node[node] (1) at (0,0) {0};
\node[node] (2) at (1,0) {1};
\node[node] (3) at (2,0) {3};
\node[node] (4) at (0,1) {4};
\node[node] (5) at (-1,1) {5};
\node[node] (6) at (-1,2) {6};
\draw[edge] (1) to[bend left] node[above] {b} (2);
\draw[edge] (2) to[bend left] node[above] {b} (3);
\draw[edge] (3) to[bend right] node[below] {a} (2);
\draw[edge] (4) to[bend left] node[below] {1-x} (2);
\draw[edge] (2) to[bend left] node[below] {a} (4);
\draw[edge] (5) to[bend left] node[below] {a} (6);
\draw[edge] (6) to[bend left] node[above] {b} (5);
\draw[edge] (5) to[bend left] node[above] {a} (4);
\end{tikzpicture}
\end{center}

**Figure 3:** Auxiliary automaton for the value 1 problem.

**Proposition 43.** Let \( x \in (0, 1) \), then the automaton \( \mathcal{A}_x \) from Figure 3 has value 1 if and only if \( x > \frac{1}{2} \).

**Proof.** The automaton \( \mathcal{A}_x \) is in fact built similarly to \( \mathcal{C} \) in Figure 2c but with probabilities \( x \) and \( 1-x \) instead of \( \frac{1}{2} \), and by putting two copies together. First notice that
\[
\mathcal{A}_x \left( \begin{array}{c} 1 \rightarrow a \rightarrow b \\ 3 \end{array} \right) = x^n, \quad \mathcal{A}_x \left( \begin{array}{c} 1 \rightarrow b \rightarrow a \\ 1 \end{array} \right) = 1 - x^n. \quad \mathcal{A}_x \left( \begin{array}{c} 4 \rightarrow a \rightarrow b \\ 6 \end{array} \right) = (1-x)^n. \quad \mathcal{A}_x \left( \begin{array}{c} 4 \rightarrow a \rightarrow b \\ 4 \end{array} \right) = 1 - (1-x)^n.
\]
The first one is immediate since the only path from 1 to 3 labelled by $a^n b$ is to stay in 1. The second one follows by stochasticity since reading $a^n b$ from 1 leads either to 1 or 3. The last two are symmetric by replacing $x$ with $1 - x$.

Let $n_1, \ldots, n_k \geq 1\text{ and } w = a^{n_1} b \cdots a^{n_k} b$, then (see Exercise 44)

$$A_x \left( 1 \xrightarrow{w} 3 \right) = 1 - \prod_{i=1}^{k} (1 - x^{n_i}) \quad \text{and} \quad A_x \left( 4 \xrightarrow{w} 6 \right) = 1 - \prod_{i=1}^{k} (1 - (1 - x)^{n_i}) \leq \sum_{i=1}^{k} (1 - x)^{n_i}. \quad \text{(3)}$$

If $x \leq \frac{1}{2}$ then $A_x \left( 1 \xrightarrow{w} 3 \right) \leq A_x \left( 4 \xrightarrow{w} 6 \right)$ and since 6 is not accepting, $w$ cannot be accepted with arbitrarily high probability. Conversely, if $x > \frac{1}{2}$ then for every $\varepsilon > 0$ we will build a sequence $(n_k)_k$,

$$\sum_{k=1}^{\infty} (1 - x)^{n_k} \leq \varepsilon \quad \text{and} \quad \sum_{k=1}^{\infty} x^{n_k} = \infty.$$

Let $c \in \mathbb{R}$ to be fixed later, and $n_k = c + \ln_x \frac{1}{\varepsilon}$. Check that $\sum_{k=1}^{\infty} x^{n_k} = x^c \sum_{k=1}^{\infty} \frac{1}{x^k} = \infty$. On the other hand, since $x \in (0, \frac{1}{2})$ and by continuity, there exists $\beta > 1$ such that $1 - x = x^\beta$. Then $\sum_{k=1}^{\infty} (1 - x)^{n_k} = \sum_{k=1}^{\infty} x^{\beta n_k} = x^{\beta c} \sum_{k=1}^{\infty} \frac{1}{x^k} \leq \varepsilon$ if we choose $c$ sufficiently small. If now consider the word $w = a^{n_1} b \cdots a^{n_k} b$ for some $k \in \mathbb{N}$ then

$$A_x(bw) = \frac{1}{2} \left( 1 - A_x \left( 4 \xrightarrow{w} 6 \right) \right) + \frac{1}{2} A_x \left( 1 \xrightarrow{w} 3 \right)$$

\[ \geq 1 - \frac{1}{2} \sum_{i=1}^{k} (1 - x)^{n_i} - \frac{1}{2} \prod_{i=1}^{k} (1 - x^{n_i}) \]

\[ \geq 1 - \frac{1}{2} \sum_{i=1}^{\infty} (1 - x)^{n_i} - \frac{1}{2} \exp \left( \sum_{i=1}^{k} \ln(1 - x^{n_i}) \right) \]

\[ \geq 1 - \frac{1}{2} \varepsilon - \frac{1}{2} \exp \left( - \sum_{i=1}^{k} x^{n_i} \right) \]

but since $\sum_{i=1}^{\infty} x^{n_i} = \infty$, we can find a $k \in \mathbb{N}$ such that $\exp \left( - \sum_{i=1}^{k} x^{n_i} \right) \leq \varepsilon$ and then $A_x(bw) \geq 1 - \varepsilon$. \hfill \Box

Exercise 44. In the proof of Proposition 43, prove (3).

Theorem 45. The value 1 problem is undecidable.

Proof. We will reduce from the strict emptiness problem with fixed cut-point $\frac{1}{2}$. Let $B$ be a probabilistic automaton over alphabet $A$, which we assume does not contain $a$ and $b$. We will now combine $A_x$ from Figure 3 and $B$. The idea is to replace the transitions in $A_x$ that involve $x$ by copies of $B$. Consider the automaton $C$ below, over alphabet $A \cup \{b, \#\}$, where the transitions coming out of $B$ are from the accepting states of $B$, and the dashed transitions coming out of $B$ are from the non-accepting states. Furthermore, the only accepting states of $C$ are 5 and 3. The notation $\forall a.a$ means that we add all transitions labelled by $a \in A$. For convenience, we haven’t added the sink state 6 and all the transitions to it for missing/invalid letters.

We now claim that $C$ has value 1 if and only if $\exists w \in A^*$ such that $B(w) > \frac{1}{2}$.

Assume there exists such a word $w$ and let $x = B(w)$. Let $\varepsilon > 0$, then $A_x$ has value 1 by Proposition 43, so there exists $n_1, \ldots, n_k$ such that $A_x(ba^{n_1} b \cdots a^{n_k} b) \geq 1 - \varepsilon$. But now observe that

$$C(ba^{n_1} b \cdots a^{n_k} b) = C(a_{n_1} b \cdots a^{n_k} b) = 1 - \varepsilon.$$

Indeed, for the left copy of $B$, after reading a word $w_x^*$, we get to state 5 with probability $B(w) = x$ and to the initial state of $B$ with the remaining probabilistic $1 - x$, thus perfectly emulating the $a$ transition of $A_x$. The right copy of $B$ is similar except that we inverted the accepting states so the transition probabilities are swapped.
Conversely, assume that $B(w) \leq \frac{1}{2}$ for every word $w \in A^*$. It is clear that any word accepted by $C$ with positive probability must be of the form $w' = u_0v_0u_1v_1\cdots u_kv_k$ where $u_i \in b^*$ and $v_i \in A^*$. Let $x = \max_{i=1}^k B(v_i)$, then $x \leq \frac{1}{2}$ by assumption and by construction, any transition $v_i\sharp$ perfectly emulates a $a$ transition of $A_x$ (but with probability $\leq x$). Thus $C(w') \leq A_x(u_0a\null u_1\cdots u_k\null a) = \text{val}(A_x)$. But $\text{val}(A_x) < 1$ by Proposition 43 since $x \leq \frac{1}{2}$, this shows that $\text{val}(C) \leq \text{val}(A_x) < 1$.

\[ \square \]

### 1.6 The value approximation problem

We have seen in the previous sections that it is undecidable to check whether a cut-point $\lambda$ is isolated, even when $\lambda = 1$. Said differently, we cannot determine if the value (which is a limit) is bigger than a particular number. However, it seems reasonable to expect that this limit is at least \emph{approximable} with small or even arbitrarily small error. A very surprising result by Condon and Lipton is that even computing an approximation with error strictly less than $\frac{1}{2}$ is impossible [CL89]. In fact, Fijalkow showed an even stronger result: in some sense, we cannot approximate it even if we allow the algorithm to be incorrect or not terminating sometimes [Fij17].

**Theorem 46.** There is no algorithm such that given a probabilistic automaton $A$,

- if $\text{val}(A) = 1$, then the algorithm outputs “yes”;
- if $\text{val}(A) \leq \frac{1}{2}$, then the algorithm outputs “no”;
- otherwise, the algorithm can output anything or not terminate.

Proof. We look at the strict problem first: if $1 > \lambda \geq \frac{1}{2}$ then this is immediate. Indeed, if the problem was decidable, it would give us an algorithm that, in particular, outputs “yes” when $\text{val}(A) = 1$ and “no” when $\text{val}(A) \leq \frac{1}{2}$, which contradicts Theorem 46. When $\lambda < \frac{1}{2}$, assume that we have an algorithm that decides whether $\text{val}(A) > \lambda$. Let $k \in \mathbb{N}$ be such that $2^k \lambda \in \left(\frac{1}{2}, 1\right)$ and consider the algorithm that given $A$, builds $B$ such that $B(w) = 2^{-k}A(w)$ and runs the algorithm on $B$. Then $\text{val}(B) = 2^{-k}\text{val}(A)$ and therefore the algorithm accepts if and only if $\text{val}(A) > 2^{-k}\lambda$. But since $2^{-k}\lambda \geq \frac{1}{2}$, we are back to the previous case where we have shown that such an algorithm cannot exists.

The non-strict problem is exactly the same except that the case distinction is on $\lambda > \frac{1}{2}$. The value 1 problem is clearly equivalent to $\text{val}(A) \geq 1$.

**Corollary 47.** The problems of deciding, given an automaton $A$ and a value $\lambda$ whether $\text{val}(A) > \lambda$ (resp. $\text{val}(A) \geq \lambda$) is undecidable. In particular, the value $1$ problem is undecidable.

Proof. Let $A$ be an automaton and $\lambda \in (0, 1)$ a cut-point, then observe that there exists $w$ such that $A(w) > \lambda$ if and only if $\text{val}(A) > \lambda$. But checking if the value of an automaton is strictly bigger than $\lambda$ is undecidable by the previous corollary. The emptiness problem reduces to the strict emptiness problem using a similar construction to that of Proposition 31, and universality is simply the emptiness of the complement.

**Corollary 48.** The emptiness, strict-emptiness and universality problems are undecidable.

Proof. Let $A$ be an automaton and $\lambda \in (0, 1)$ a cut-point, then observe that there exists $w$ such that $A(w) > \lambda$ if and only if $\text{val}(A) > \lambda$. But checking if the value of an automaton is strictly bigger than $\lambda$ is undecidable by the previous corollary. The emptiness problem reduces to the strict emptiness problem using a similar construction to that of Proposition 31, and universality is simply the emptiness of the complement.

**Corollary 49.** The isolation problem is undecidable.

Proof. Given an automaton $A$ and a cut-point $\lambda$, we can construct, by multiplication and convex combinations, an automaton $B$ such that $B(w) = f(A(w))$ where $f(x) = (1 - \frac{x}{2})(\frac{1-\lambda}{2} + \frac{x}{2})$. By stochasticity, $f([0, 1]) \subseteq [0, 1]$ and one can check that its maximum is $y = \frac{1}{2}(1 - \frac{1}{2})(1 - \frac{2\lambda}{3})$ and it is attained only for $x = \lambda$. By continuity of $f$, it follows that $\lambda$ is isolated for $A$ if and only if $\text{val}(B) < y$ if and only if $\text{val}(B^*) > 1 - y$. But the latter is undecidable, therefore the isolation problem is (note that $y$ is clearly rational when $\lambda$ is rational).

We start the proof by studing the so-called expanding automaton $B_x$ on Figure 4a, where $x \in [0, 1]$ is arbitrary. We will later “replace” $x$ by an automaton to obtain Figure 4b.

**Proposition 50.** If $x > \frac{1}{2}$ then $\text{val}(B_x) = 1$. Specifically, $B_x((\text{check} \cdot \text{sim}^n)^2n) \to 1$ as $n \to \infty$.

Proof. We first study the behavior of $\text{check} \cdot \text{sim}^n$ in the automaton. Observe that for any $n \in \mathbb{N}$,

$B_x\left(p \text{\check{c}heck sim}^n \rightarrow L\right) = \frac{1}{2}x^n \quad \text{and} \quad B_x\left(p \text{\check{c}heck sim}^n \rightarrow R\right) = \frac{1}{2}(1-x)^n$.

Indeed, $L$ and $R$ are symmetric in the automaton (by replacing $x$ by $1-x$) so we prove it for $L$. After reading $\text{check}$, the automaton can be in state $L$ or $R$. But there is no path labelled by $\text{sim}^*$ from $R$ to $L$. Once in $L$, reading $\text{sim}$ can
make the automaton stay in \( L \) or go back to \( p \). But again there is no path labelled by \( \text{sim}^* \) from \( p \) to \( L \). Therefore the only path from \( p \) to \( L \) with positive probability goes to \( L \) first and then stays in \( L \). In other words,

\[
B_x \left( p \xrightarrow{\text{check \cdot sim}^*} L \right) B_x \left( L \xrightarrow{\text{sim}} L \right)^n = \frac{1}{2} x^n.
\]

On the other hand, by stochasticity,

\[
B_x \left( p \xrightarrow{\text{check \cdot sim}^*} p \right) = 1 - B_x \left( p \xrightarrow{\text{check \cdot sim}^*} L \right) - B_x \left( p \xrightarrow{\text{check \cdot sim}^*} R \right) = 1 - \frac{1}{2} x^n - \frac{1}{2} (1 - x)^n.
\]

Observe that when reading \( \text{check \cdot sim}^* \) from \( L \), the word is accepted with probability 1; and when reading \( \text{check \cdot sim}^* \) from \( R \), the word is rejected, i.e. accepted with probability 0.

We will now study the effect of reading \( \text{check \cdot sim}^* \) many \( (N) \) times, by induction on \( N \). After reading \( (\text{check \cdot sim}^*)^N \), the automaton can be in any state, but the only states that lead to an accepting state when reading \( \text{check \cdot sim} \) are \( L \) and \( q_L \). When reading \( \text{check \cdot sim}^n \) from either, it is accepted with probability 1, therefore

\begin{equation}
B_x((\text{check \cdot sim}^*)^{N+1}) = B_x \left( p \xrightarrow{\text{check \cdot sim}^N} q_L \right) + B_x \left( p \xrightarrow{\text{check \cdot sim}^N} L \right). \tag{4}
\end{equation}

But the only state from which \( L \) is reachable by reading \( \text{check \cdot sim} \) is \( p \). Therefore

\begin{equation}
B_x \left( p \xrightarrow{(\text{check \cdot sim}^*)^N} L \right) = B_x \left( p \xrightarrow{\text{check \cdot sim}^N} p \right) B_x \left( p \xrightarrow{\text{check \cdot sim}^N} L \right). \tag{5}
\end{equation}

Similarly, the only state from which \( p \) is reachable by reading \( \text{check \cdot sim}^n \) is \( p \). Therefore

\begin{equation}
B_x \left( p \xrightarrow{(\text{check \cdot sim}^*)^N} p \right) = B_x \left( p \xrightarrow{\text{check \cdot sim}^N} p \right)^N. \tag{6}
\end{equation}

Let \( p_n = \frac{1}{2} x^n \) and \( q_n = \frac{1}{2} (1 - x)^n \) be the probabilities that we computed at the beginning. Then

\[
B_x((\text{check \cdot sim}^*)^N) = \sum_{i=1}^{N-1} B_x \left( p \xrightarrow{(\text{check \cdot sim}^*)^i} L \right) \quad \text{by (4)}
\]

\[
= \sum_{i=1}^{N-1} B_x \left( p \xrightarrow{(\text{check \cdot sim}^*)^{i-1}} p \right) B_x \left( p \xrightarrow{\text{check \cdot sim}^N} L \right) \quad \text{by (5)}
\]

\[
= \sum_{i=1}^{N-1} (1 - p_n - q_n)^{i-1} p_n \quad \text{by (6)}
\]

\[
= p_n \sum_{i=0}^{N-2} (1 - p_n - q_n)^i
\]

\[
= p_n \frac{1 - (1 - p_n - q_n)^{N-1}}{1 - (1 - p_n - q_n)}
\]

\[
= \frac{p_n}{p_n + q_n} \left( 1 - (1 - p_n - q_n)^{N-1} \right)
\]

\[
= 1 + \frac{q_n}{p_n} \left( 1 - (1 - p_n - q_n)^{N-1} \right).
\]

We now let \( N = 2^n \) and assume that \( x > \frac{1}{2} \). Then \( \frac{1 - x}{x} < 1 \) so \( 2^n \cdot (\frac{1 - x}{x})^n \to 0 \) as \( n \to \infty \). Similarly, \( 1 - p_n - q_n < 1 - \frac{1}{2} x^n \leq 1 - \frac{1}{2} \) for all \( n \). Therefore \( (1 - p_n - q_n)^{N-1} \to 0 \) as \( n \to \infty \) and \( B_x((\text{check \cdot sim}^*)^{2^n}) \to 1 \) as \( n \to \infty \). By definition, \( \text{val}(B_x) \geq B_x((\text{check \cdot sim}^*)^{2^n}) \) for all \( n \in \mathbb{N} \) so \( \text{val}(B_x) = 1 \).

We can now show the main result.

**Proof of Theorem 46.** Let \( C \) be any probabilistic automaton on some alphabet \( A \), which we assume (without loss of generality) to only have one initial state \( q_0 \) that is not accepting, and consider automaton \( D \) on alphabet \( \Sigma = A \cup \{ \text{check, end} \} \) from Figure 4b. The transitions coming out of \( C \) are from the accepting states of \( C \), the \( \text{dashed} \) transitions coming out of \( C \) are from the non-accepting the states, the \( \text{dotted} \) transitions coming out of \( C \) are only from \( q_0 \). We rename the state \( q_0 \) to \( L \) in \( C_L \) and to \( R \) in \( C_R \). Let \( w \in A^* \), when reading \( w \cdot \text{end} \):

- from \( p, q_L, q_R \): we stay in this state with probability 1,
• from $L$: we stay in $L$ with probability $x$ and go to $p$ with probability $1 - x$,

• from $R$: we stay in $R$ with probability $1 - x$ and go to $p$ with probability $x$.

We observe that this is the same transition table as $\sim$ in $B_x$ when $x = C(w)$. We have shown in Proposition 50 that if $x > \frac{1}{2}$ then $\text{val}(B_x) = 1$. Specifically, $B_x((\text{check} \cdot \sim^n )^2) \rightarrow 1$ as $n \rightarrow \infty$. But we have observed that $B_x((\text{check} \cdot \sim^n )^2) = D((\text{check} \cdot (w \cdot \text{end})^n )^2)$ since the transition table is the same for $\sim$ (and is obviously the same for other letters). Therefore

$$\text{val}(D) = 1 \text{ if there exists a word } w \text{ such that } C(w) > \frac{1}{2} \quad (7)$$

Conversely, let $w \in \Sigma^*$, we will show that

$$D \left( p \xrightarrow{w} q_L \right) \leq D \left( p \xrightarrow{w} q_R \right) \quad \text{ and } \quad D \left( p \xrightarrow{w} L \right) \leq D \left( p \xrightarrow{w} R \right)$$

by induction by considering the following cases: $w = w' \cdot \text{check} \cdot A^* \cdot \text{end}$, $w = w' \cdot \text{check} \cdot A^*$ and $w \in (A \cup \{\text{end}\})^*$. This case distinction is exhaustive since if $w \in (A \cup \{\text{end}\})^*$, then the automaton is always in $p$, thus all other probabilities are 0 and the inequalities hold. Note that this covers the initial induction step ($w = \varepsilon$). Otherwise, $w$ must contain at least one $\text{check}$ and it either finishes by $\text{end}$ or by a (possibly empty) word in $A^*$.

• If $w = w' \cdot \text{check} \cdot u$ with $u \in A^*$ then after reading $w' \cdot \text{check}$ the automaton must be in state $L, q_L, R$ or $q_R$. Furthermore, for any $s, t \in \{L, R, q_L, q_R\}$, if $s \neq t$ then there is no transition from $s$ to $t$ labelled by $u$, i.e. $D \left( s \xrightarrow{u} t \right) = 0$. Therefore, for any $s \in \{L, R, q_L, q_R\}$, $D \left( p \xrightarrow{w'} s \right) = D \left( p \xrightarrow{w' \cdot \text{check}} s \right) \cdot D \left( s \xrightarrow{u} s \right)$. Therefore

$$D \left( p \xrightarrow{w} q_L \right) = D \left( p \xrightarrow{w' \cdot \text{check}} q_L \right)$$

$$= D \left( p \xrightarrow{w'} L \right)$$

$$\leq D \left( p \xrightarrow{w'} R \right)$$

by induction

$$D \left( p \xrightarrow{w} q_R \right)$$

by a symmetric reasoning.

Similarly,

$$D \left( p \xrightarrow{w} L \right) = D \left( p \xrightarrow{w' \cdot \text{check}} L \right) \cdot C \left( q_0 \xrightarrow{u} q_0 \right)$$

$$= D \left( p \xrightarrow{w'} p \right) \cdot C \left( q_0 \xrightarrow{u} q_0 \right)$$

$$= D \left( p \xrightarrow{w} R \right)$$

by a symmetric reasoning.

• If $w = w' \cdot \text{check} \cdot u \cdot \text{end}$ with $u \in A^*$ then after reading $w' \cdot \text{check}$ the automaton must be in state $L, q_L, R$ or $q_R$. The analysis for $q_L$ and $q_R$ is the same because there are no transitions from $L$ or $R$ to $q_L$ or $q_R$ labelled by $u \cdot \text{end}$. The analysis for $L$ and $R$ is a bit different: note that for $L$ to be reachable $w$, the automaton must be in state $p$ when reading $\text{check} \cdot u \cdot \text{end}$ and similarly for $R$. Therefore

$$D \left( p \xrightarrow{w} L \right) = D \left( p \xrightarrow{w'} p \right) \cdot D \left( p \xrightarrow{\text{check} \cdot u \cdot \text{end}} L \right)$$

$$= D \left( p \xrightarrow{w'} p \right) \cdot D \left( p \xrightarrow{\text{check} \cdot \text{end}} L \right) \cdot D \left( L \xrightarrow{u \cdot \text{end}} L \right)$$

$$= \frac{1}{2} D \left( p \xrightarrow{w'} p \right) \cdot C(u)$$

$$\leq \frac{1}{2} D \left( p \xrightarrow{w'} p \right) \cdot (1 - C(u))$$

$$= D \left( p \xrightarrow{w'} p \right) \cdot D \left( \text{check} \xrightarrow{} R \right) \cdot D \left( R \xrightarrow{u \cdot \text{end}} R \right)$$

$$= D \left( p \xrightarrow{w'} p \right) \cdot D \left( \text{check} \cdot u \cdot \text{end} \xrightarrow{} R \right)$$

$$= D \left( p \xrightarrow{w} R \right)$$

by the above remark.
Markov chains and linear dynamical systems

A Markov chain is a particular case of probabilistic automata where the alphabet is unary. In this case, we can simplify the presentation and describe a Markov chain in dimension $d$ by a tuple $\mathcal{M} = \langle S, A, T \rangle$ where

- $S \in [0,1]^{1 \times d}$ is stochastic (row) vector of initial probabilities,
- $T \in \{0,1\}^{d \times 1}$ is a 0–1 (column) vector of accepting states,
- $A \in [0,1]^{1 \times d}$ is a stochastic matrix of transition probabilities.

Similarly to probabilistic automata, we usually assume that initial probabilities and transition probabilities are rational numbers. In the case of Markov chains, there is a unique probability of acceptance for every length. It is given for every $n \in \mathbb{N}$ by

$\mathcal{M}(n) = SA^nT$.

More generally, we will consider linear dynamical systems (LDS) $\langle S, A, T \rangle$ where we lift the restriction that $I$ and $A$ be stochastic. In particular, the values of a LDS do not need to be within $[0,1]$.

Example 51. Figure 5a illustrates an hypothetical stock market that can exhibit three behaviors during a week: bull, bear or stagnant. For example, following a bull week, the market has 90% chances of being bull the next week but it will become bear with a 7.5% probability. If we start from an initial distribution over the three states and put it in a vector $I$, and let $A$ be the transition matrix, then $SA^n$ gives the probability distribution over the three states after $n$ weeks. We can thus analyse the long-term behavior of the system. For example if we take $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ then $SA^nT$ gives the probability of being in a certain state (say bull) after $n$ weeks. The emptiness problem now becomes: does there exists $n$ such that $SA^nT \geq \lambda$, in other words, is there any week where the probability of the market being bull is higher than $\lambda$.

Example 52. Figure 5b illustrates a simplified model for the dynamics of a frog population. Frogs have three life stages: egg, tadpole and adult. Every year, 50% of the eggs survive to become tadpoles, 30% of the tadpoles become adults and every pair of adults produces 120 eggs and dies. The corresponding transition matrix, also known as the Leslie matrix, is

$A = \begin{bmatrix} 0 & 0 & 60 \\ 0.5 & 0 & 0 \\ 0 & 0.3 & 0 \end{bmatrix}.$
Starting from an initial state of the pond, for example 50 eggs, 20 tadpoles and 2 adults, we can get the state of the population after \( n \) years by computing \( SA^n \) where \( S = \begin{bmatrix} 50 & 20 \\ 2 & 2 \end{bmatrix} \). We can study the long-term behavior of this system, for example the total population size is given by \( SA^T \) where \( T = \begin{bmatrix} 1 & 1 \end{bmatrix}^T \).

2.1 Linear recurrent sequences

An alternative point of view is to consider the sequence \( (u_n)_{n \in \mathbb{N}} \) given by \( u_n = M(n) \). A useful property of this sequence is that it is linear. Formally, a linear recurrent sequence \( (\text{LRS}) \) of order \( k \) is any sequence \( (u_n)_{n \in \mathbb{N}} \) that satisfies the recurrence relation

\[
u_{n+k} = a_{k-1}u_{n+k-1} + \cdots + a_0u_n
\]

for all \( n \in \mathbb{N} \), for some numbers \( a_0, \ldots, a_{k-1} \in \mathbb{R} \). When all numbers \( u_n \) and \( a_i \) are rational, we say that it is a rational LRS, and if all numbers \( u_n \) and \( a_i \) are integers, then it is an integer LRS. There is a strong connection between LRS and matrix powers that comes from linear algebra.

**Theorem 53** (Cayley–Hamilton). Let \( A \in \mathbb{R}^{d \times d} \) be a matrix and let \( p(\lambda) = \det(\lambda I_d - A) \) be its characteristic polynomial, then \( p(A) = 0 \). In particular, \( A^d \) is a linear combination of \( I_d, A, \ldots, A^{d-1} \).

**Proof.** We admit the proof and simply show how the last statement follows from \( p(A) = 0 \). Indeed, \( p(\lambda) \) is a determinant of \( d \times d \) matrix, thus it is a polynomial of degree \( d \) in \( \lambda \). Furthermore, it is not hard to see that \( p \) is monic, i.e. \( p(\lambda) = \lambda^d + q(\lambda) \) where \( q \) has degree at most \( d - 1 \). Therefore, \( p(A) = 0 \) implies that \( A^d = -q(A) = \sum_{i=0}^{d-1} a_i A^i \) where the \( a_i \) are the coefficients of \( q \).

**Proposition 54.** Let \( d \in \mathbb{N} \), let \( S \in \mathbb{Q}^{1 \times d}, A \in \mathbb{Q}^{d \times d} \) and \( T \in \{0,1\}^{d \times 1} \). Then the sequence \((SA^n T)_{n \in \mathbb{N}} \) is a rational LRS of order \( d \). Furthermore, if all entries of \( I \) and \( A \) are integers, then it is an integer LRS. In particular, if \( M \) is a Markov chain, then \((M(n))_{n \in \mathbb{N}} \) is a rational LRS. Conversely, if \((u_n)_{n \in \mathbb{N}} \) is rational LRS of order \( d \), then there exists a LDS \((S,A,T)\) of dimension \( d \) such that \( u_n = SA^n T \) for all \( n \in \mathbb{N} \). Furthermore, if \((u_n)_{n \in \mathbb{N}} \) is an integer LRS then \( S, A \) and \( T \) have integer coefficients.

**Proof.** By Cayley–Hamilton theorem, \( A^d \) is a linear combination of \( I_d, A, \ldots, A^{d-1} \) so we can find \( a_0, \ldots, a_{d-1} \in \mathbb{Q} \) such that

\[
A^d = \sum_{i=0}^{d-1} a_i A^i.
\]

For every \( n \in \mathbb{N} \), let \( u_n = SA^{n+d} T \), then we have that

\[
u_{n+d} = SA^{n+d} T = SA^n A^d T = SA^n \left( \sum_{i=0}^{d-1} a_i A^i \right) T = \sum_{i=0}^{d-1} a_i S A^{n+i} T = \sum_{i=0}^{d-1} a_i u_{n+i}.
\]

Thus \((u_n)_{n} \) is a LRS. If all entries of \( I \) and \( A \) are rational, then the characteristic polynomial \( p \) of \( A \) has rational entries thus the coefficients \( a_i \) are rationals. Similarly if \( I \) and \( A \) are rational, then the coefficients of \( p \) are integers.
Conversely, if \((u_n)_n\) is a LRS of order \(d\), let \(a_0, \ldots, a_{d-1}\) be the coefficients of the recurrence relation. Consider the matrices
\[
S = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{d-1} \end{bmatrix}.
\]
Then we check that for every \(n \in \mathbb{N}\),
\[
A \begin{bmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+d-1} \end{bmatrix} = \begin{bmatrix} u_{n+1} \\ \vdots \\ u_{n+d} \end{bmatrix} \quad \text{and thus} \quad A^n T = \begin{bmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+d-1} \end{bmatrix}.
\]
follows by induction. This implies that \(SA^n T = u_n\) and concludes. \(\square\)

**Remark 55.** The proof of Proposition 54 could give the impression that any Markov chain or LDS \((S, A, T)\) verifies equation (9), i.e. it shifts consecutive terms by one. This is not the case in general, see Exercise 56.

**Exercise 56.** Consider the following two LDS \((S, A_1, T)\) and \((S, A_2, T)\):
\[
S = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad T = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.
\]
By following the proof of Proposition 54, let \(u_n = SA^n T\) find the recurrence relation (of order 2) satisfied by \(u_n\), find \(u_0\) and \(u_1\) and give an explicit expression for \(u_n\). Find an explicit expression for \(B^n\) and show that \(u_n = SB^n T\). Then prove that
\[
A^n \begin{bmatrix} u_n \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} u_{n+1} \\ u_n + d - 1 \end{bmatrix} \quad \text{but} \quad B^n \begin{bmatrix} u_n \\ u_{n+1} \end{bmatrix} \neq \begin{bmatrix} u_{n+1} \\ u_n + d - 1 \end{bmatrix}.
\]

**Proposition 57.** Let \(\lambda \in \mathbb{Q}\), \((u_n)_n\) and \((v_n)_n\) be two rational LRS. Then \((\lambda u_n)_n\), \((u_n + v_n)_n\), \((u_nv_n)_n\).

**Proof.** For Markov chains, this is a special case of Section 1.2, and in fact the same proofs works for non-stochastic systems as well. We redo the proof for completeness.

The first item is trivial since it satisfies \(u_{n+1} = u_n\). Let \((u_n)_n\) and \((v_n)_n\) be two LRS of order \(d\) (we can always increase the order artificially) and let \(a_0, \ldots, a_{d-1}\) and \(b_0, \ldots, b_{d-1}\) be the coefficients of the recurrence relation. Let \(w_n = \lambda u_n\), then
\[
w_{n+d-1} = \lambda u_{n+d-1} = \lambda \sum_{i=0}^{d-1} a_i u_{n+i} = \sum_{i=0}^{d-1} a_i \lambda u_{n+i} = \sum_{i=0}^{d-1} a_i w_{n+i}
\]
thus \((w_n)_n\) is a LRS. By Proposition 54, there exists \(S_1, S_2, A_1, A_2, T_1\) and \(T_2\) such that \(u_n = S_1 A_1^n T_1\) and \(v_n = S_2 A_2^n T_2\). Consider
\[
\hat{S} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} A_1 \\ 0 \\ A_2 \end{bmatrix}, \quad \hat{T} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}.
\]
Then we have that
\[
\hat{S} \hat{A}^n \hat{T} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \begin{bmatrix} A_1^n \\ 0 \\ A_2^n \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = S_1 A_1^n T_1 + S_2 A_2^n T_2 = u_n + v_n.
\]
Thus by Proposition 54, \((u_n + v_n)_n\) is a LRS. Similarly, consider
\[
\hat{I} = S_1 \otimes S_2, \quad \hat{A} = A_1 \otimes A_2, \quad \hat{T} = T_1 \otimes T_2
\]
where \(\otimes\) denotes the Kronecker product (see proof of Lemma 22). Then by the mixed-product property,
\[
\hat{S} \hat{A}^n \hat{T} = (S_1 \otimes S_2)(A_1 \otimes A_2)^n(T_1 \otimes T_2) = (S_1 A_1^n T_1) \otimes (S_2 A_2^n T_2) = u_n v_n.
\]
Thus by Proposition 54, \((u_nv_n)_n\) is a LRS. \(\square\)

Another interesting feature of LRS is that we can provide an explicit expression for its general term. Let \((u_n)_n\) be a LRS and let \(a_0, \ldots, a_{d-1}\) be the coefficients of its recurrence relation. We define the characteristic polynomial of the sequence to be
\[
p(x) = x^d - a_1 x^{d-1} \cdots a_{d-1} x - a_d.
\]
**Proposition 58.** Let \((u_n)_n\) be a LRS and \(p\) its characteristic polynomial. Let \(\lambda_1, \ldots, \lambda_d\) be the (possibly repeated) (complex) roots of \(p\). Then there are univariate polynomials \(A_1, \ldots, A_d\) of degree at most \(d\) such that

\[
    u_n = A_1(n)\lambda_1^n + \cdots + A_d(n)\lambda_d^n.
\]

In particular, \((u_n)_n\) is linear combination of the sequences \(n^k\lambda_i^n\) for \(i \in \{1, \ldots, d\}\) and \(0 \leq k < d\). Furthermore, all the coefficients that appear in the \(A_i\) are algebraic numbers\(^3\). Conversely, any sequence of this form is a LRS.

**Proof.** Put \(A\) in Jordan Normal Form (see Proposition 59) below, then \(A = PJP^{-1}\) where \(J = \text{diag}(J_1, \ldots, J_k)\). It follows that \(A^n = PJ^n P^{-1}\) and \(J^n = \text{diag}(J_1^n, \ldots, J_k^n)\). It is easy to check by induction that a block \(J_i\) of dimension \(k\) satisfies

\[
    J_i^n = \begin{bmatrix}
        (n)\lambda_i^n & (n-1)\lambda_i^{n-1} & \cdots & (1)\lambda_i^1 \\
        \vdots & \ddots & \vdots \\
        (n)\lambda_i^n & \cdots & (1)\lambda_i^1 \\
        \lambda_i^n & \cdots & \lambda_i^n
    \end{bmatrix}
\]

and therefore the entries of \(J_i^n\) are a linear combination of \(n^k\lambda_i^n\). Putting everything together, we get the result. \(\square\)

**Proposition 59 (Jordan Normal Form (JNF)).** Let \(A \in \mathbb{R}^{d \times d}\) be a matrix, then there exists an invertible matrix \(P\) and block diagonal matrix \(J = \text{diag}(J_1, \ldots, J_k)\) such that \(A = PJP^{-1}\) where \(J_i\) is a Jordan block of the form

\[
    J_i = \begin{bmatrix}
        \lambda_i & 1 \\
        \vdots & \ddots \\
        \lambda_i & 1
    \end{bmatrix}
\]

where \(\lambda_1, \ldots, \lambda_k\) are the (possibly repeated) eigenvalues of \(A\).

To summarize, we started with Markov chains that we generalized to linear dynamical systems. We then showed that the following objects are equivalent:

- linear dynamical systems,
- linear recurrent sequence,
- exponential polynomials: expressions of the form \((10)\).

This equivalence is important because it shows that LRS are a universal object in some sense, they appear naturally in many contexts and it gives more tools to solve problems.

### 2.2 Decision problems

Recall that in Section 1.3, we looked at the emptiness problem for probabilistic automata and showed that it is undecidable. It is clear that the proof does not apply anymore because we used binary expansion to encode words, something which is impossible with a unary alphabet. In fact, the problem becomes impossible with a unary alphabet. In fact, the problem becomes much simpler. Indeed, fix \(\lambda \in (0, 1)\), then the emptiness problem for Markov chain becomes: decide whether there exists \(n \in \mathbb{N}\) such that \(SA^nT > \lambda\). Note in particular that this is a “deterministic” problem: there are no words to choose, we just need to check if a LRS contains an element bigger than \(\lambda\). For this purpose, we introduce the following two problems (the names are not universally):

**Problem 60 (Markov Reachability/Equality).** Given a Markov chain \(\langle S, A, T \rangle\) and a threshold \(\lambda \in \mathbb{Q}\), decide whether \(SA^nT = \lambda\) for some \(n\).

**Problem 61 (Markov inequality).** Given a Markov chain \(\langle S, A, T \rangle\) and a threshold \(\lambda \in \mathbb{Q}\), decide whether \(SA^nT \geq \lambda\) for all \(n\) version.

Note that the Markov inequality problem naturally comes in two flavors, depending on whether the inequality is strict or not. It is clear that the Markov inequality problem is equivalent (in terms of decidability) to the emptiness problem since \(\exists n.SA^nT > \lambda\) if and only if \(\forall n.SA^nT \leq \lambda\) if and only if \(\forall n.(1 - SA^nT) \geq 1 - \lambda\) and \(1 - SA^nT\) is also a Markov chain.

While the Markov reachability problem hasn’t necessarily received a lot of attention, the following well-known problems for integer LRS have been studied extensively.

\(^3\)Algebraic numbers are roots of polynomials with integer (or rational coefficients). For example \(x = \sqrt{2}\) is algebraic because \(x^2 - 2 = 0\).
Problem 62 (Skolem). Given a LRS \((u_n)_n\), decide whether it has a zero, i.e., whether \(u_n = 0\) for some \(n \in \mathbb{N}\).

Problem 63 (Positivity). Given a LRS \((u_n)_n\), decide whether it is positive, i.e., whether \(u_n > 0\) for all \(n \in \mathbb{N}\).

Note that the positivity problem also naturally comes in two flavors, depending on whether the inequality is strict or not. It is clear that the positivity problem is harder than the Skolem problem since we can reduce the latter to the former. On the other hand, the Skolem problem has now been open for more than 70 years [OW12]! In particular, the Skolem problem is not known to be either decidable or undecidable.

Remark 64. The Skolem and positivity are classically defined with a threshold of 0. This is without loss of generality since for any \(\lambda \in \mathbb{Q}\), \(u_n = \lambda\) if and only if \(u_n - \lambda = 0\) and \((u_n - \lambda)_n\) is a LRS.

Exercise 65. Some authors define the Skolem or Markov reachability problem as follows: given a matrix \(A \in \mathbb{Q}^{d \times d}\), decide whether \((M^n)_{1,2} = 0\) for some \(n\). Show that the two formulations are equivalent.

It is clear that the Markov reachability and inequality problems are particular cases of the Skolem and positivity problems for rational LRS. Nevertheless, one could hope that the stochastic aspect could make the problem easier. We will show that this is unfortunately not the case. The reduction follows [Aks+15] and uses the following intermediate problem.

Problem A. Given a stochastic matrix \(A \in \mathbb{Q}^{d \times d}\) and a vector \(y \in \{0,1,2\}^d\), decide whether \(eA^ny = 1\) where \(e = [1\ 0 \ldots 0]\).

Proposition 66. The Skolem problem for rational LRS reduces to Problem A.

Proof. Let \(A \in \mathbb{Z}^{d \times d}\) be an instance of the Skolem problem. Without loss of generality (see Exercise 65), we are trying to decide whether \((A^n)_{1,2} = 0\) for some \(n\). We will construct a stochastic matrix \(\tilde{P}\) and vector \(\tilde{v} \in \{0,1,2\}^{2k+1}\) such that for all \(n\), \(A^n_{1,2} = 0\) if and only if \(e\tilde{P}^n\tilde{v} = 0\) where \(e = [1\ 0 \ldots 0]\).

The first step consists in separating the positive and negative values in \(A\), since a stochastic matrix can only have nonnegative entries. Let \(A^+ + A^-\) be nonnegative matrices defined by \(A^+_{ij} = \max(0,A_{ij})\) and \(A^-_{ij} = \max(0,-A_{ij})\), then \(A = A^+ - A^-\). Now define
\[
e = [1\ 0 \ldots 0], \quad P = \begin{bmatrix} A^+ & A^- \\ A^- & A^+ \end{bmatrix}, \quad v = \begin{bmatrix} x \\ -x \end{bmatrix}, \quad x = [0\ 1\ 0 \ldots 0]^t.
\]

One checks that \(e\tilde{P}^nv = e(A^+ - A^-)^nx = A^n_{1,2}\) by using that \(\varphi : \begin{bmatrix} X \ Y \\ Y \ X \end{bmatrix} \mapsto X - Y\) is a homomorphism from the ring of \(2k \times 2k\) symmetric matrices to \(k \times k\) matrices.

The second step is to rescale the matrix to make it stochastic\(^4\), now that it only has nonnegative entries. Let \(s \in \mathbb{Q}\) such that \(sP\) is substochastic and define
\[
\tilde{e} = [e\ 0], \quad P = \begin{bmatrix} sP & 1 - sP1 \\ 0 & 1 \end{bmatrix}, \quad \tilde{v} = \begin{bmatrix} 1 + v \\ 1 \end{bmatrix}, \quad \text{where} \ 1 = [1\ \ldots\ 1]^t.
\]

First it is clear that \(\tilde{e}\) is stochastic since it contains a single 1, and the entries of \(\tilde{v}\) are in \(\{0,1,2\}\) since the entries of \(v\) are in \(\{-1,0,1\}\). Moreover, \(\tilde{P}\) is stochastic since on row \(i\), the last entry is \(1 - (sP1)_i = 1 - \sum_j sP_{ij}\), i.e. the remainder to make it stochastic. Then we have that
\[
\tilde{e}\tilde{P}^nv = \begin{bmatrix} e\ 0 \end{bmatrix} \begin{bmatrix} (sP)^n & 1 - (sP)^n1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 + v \\ 1 \end{bmatrix} = [e\ 0] \begin{bmatrix} (sP)^nv + 1 \\ 1 \end{bmatrix} = e(sP)^nv + e1 = s^nA^n_{1,2} + 1.
\]

Therefore, \(\tilde{e}\tilde{P}^nv = 1\) if and only if \(A^n_{1,2} = 0\).

Proposition 67. Problem A reduces to the Markov reachability problem with threshold \(\frac{1}{2}\).

Proof. Let \(e = [1\ 0 \ldots 0]\), \(A \in \mathbb{Q}^{d \times d}\) stochastic and \(y \in \{0,1,2\}^k\) be an instance of Problem A. We will build a \((k+5) \times (k+5)\) stochastic matrix \(P\) such that \(eA^ny = 1\) if and only if \(SP^{n+2}T = \frac{1}{2}\) for some \(S\) and \(T\), which is an instance of the Markov reachability problem.

First, we need to put \(y\) in the matrix itself since we cannot have a value of 2 in the vector \(T\) and ensure that the resulting matrix is stochastic. Define
\[
s = [e\ 0\ 0], \quad B = \begin{bmatrix} \frac{1}{2}A & \frac{1}{2}y & 1 - \frac{1}{2}(A1 + y) \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad t = [0\ 1\ 0], \quad 1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.
\]

\(^4\)We have already seen this trick for probabilistic automata, but in the case at hand, we have more constraints to satisfy on the vectors.
Note that $s$ is stochastic and $B$ is stochastic since $A$ is stochastic and $y_i \leq 2$ thus each line has sum 1 and $1 - \frac{1}{4} A 1 - \frac{1}{2} y \geq 0$. Then check (using that $A 1 = 1$ since $A$ is stochastic) that

$$sB^nt = \begin{bmatrix} e & 0 & 0 \end{bmatrix} \begin{bmatrix} (\frac{1}{4}A)^n & 1 - \frac{1}{4}(A^n 1 - y) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 4^{-n} e A^{n-1} y.$$

Next, we will use the following automaton, which is essentially the same as in the proof of Proposition 31, to “compensate” for the $4^{-n}$ factor.

![Automaton](image)

Formally, define

$$S = [0 \ldots 0 1 0 0], \quad P = \begin{bmatrix} B & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} t \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

It is immediate that $S$ and $P$ are stochastic since $s$ are $B$ are stochastic, and $T$ only has \{0, 1\} entries. It can be checked by induction, or by reasoning on the automaton that

$$SP^nT = S \begin{bmatrix} B^n & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2(1 - 4^{-1}) \\ 0 & 0 & 4^{-1} & 1 - 4^{-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/2(sB^{n-1}t + 1 - 4^{-1}) \\ 1 - 4^n \\ 1 \end{bmatrix} = \frac{1}{2}(sB^{n-1}t + 1 - 4^{-1}).$$

It follows that

$$SP^nT = \frac{1}{2} \iff sB^{n-1}t + 1 - 4^{-1} = 1 \iff 4^{1-n} e A^{n-2} y + 1 - 4^{-1} = 1 \iff e A^{n-2} y = 1.$$

We can now show the main result of this section.

**Theorem 68.** The following problem are interreducible$^5$:

- the Skolem problem for integer LRS,
- the Skolem problem for rational LRS,
- the Markov reachability problem, even for fixed threshold.

*Proof.* The Skolem problem for integer LRS and the Markov reachability problem are particular case of the Skolem problem for rational LRS. By Proposition 66 and Proposition 67, we have that the Skolem problem for rational LRS is reducible to the Markov reachability problem. Finally the Skolem problem for rational LRS is easily reducible to the Skolem problem for integer LRS: let $(u_n)_n$ be a rational LRS, by Proposition 54, write $u_n = SA^nT$ for some rational $S, A, T$. Then there exists $m \in \mathbb{N}$ such that $mS, mA$ and $mT$ have integer coefficients. But clearly, $v_n = (mS)(mA)^n(mT) = m^{n+2} u_n$ thus the Skolem problem for $(u_n)_n$ is equivalent to the Skolem problem for $(v_n)_n$, but the latter is an integer LRS by Proposition 54.

**Theorem 69.** The following problem are interreducible:

- the positivity problem for integer LRS,
- the strict positivity problem for integer LRS,
- the positivity problem for rational LRS,
- the strict positivity problem for rational LRS,
- the Markov reachability problem, even for fixed threshold.

$^5$This means there are both reducible to each other. In particular their (non-)decidability is equivalent.
• the strict Markov reachability problem, even for fixed threshold.

Proof. It is straightforward to check that the proof of Theorem 68 also shows that all the non-strict problems are interreducible, and that all the strict problem are interreducible. It remains to see that a strict problem is interreducible with a non-strict one. This is the case for the integer LRS.

Let \((u_n)\) be an integer LRS, then \(u_n \geq 0\) if and only if \(u_n + 1 > 0\) and \(u_n + 1\) is an integer LRS. Thus the non-strict positivity problem reduces to the strict one. Conversely, \(u_n > 0\) if and only \(u_n \geq 1\) thus the strict positivity problems reduces to the non-strict one. \(\Box\)

2.3 Skolem–Mahler–Lech theorem

We will now see one of the most famous results on the Skolem problem, that gives the structure of the set of zeroes of a LRS. We will follow a particularly simple proof that does not require too much number theory [Han86]. A set \(A \subseteq \mathbb{N}\) is called:

• periodic if there exists \(r\) such that \(q \in A\) if and only if \(q + r \in A\) for all \(q \in \mathbb{N}\).
• ultimately periodic if there exists \(q_0\) and \(r\) such that \(q \in A\) if and only if \(q + r \in A\) for all \(q \geq q_0\).
• quasi-periodic if it the union of a finite set and a periodic set.

Exercise 70. Show that \(A\) is periodic of period \(r\) if and only if there exists a finite set \(P \subseteq \{0, \ldots, r-1\}\) such that 
\[ A = \bigcup_{p \in P} (p + r\mathbb{N}) \]
Show that \(A\) is ultimately periodic if and only if there exists \(r \in \mathbb{N}\) and two finite sets \(F, P\) such that 
\[ Z = F \cup \bigcup_{p \in P} (p + r\mathbb{N}) \]

Lemma 71. Let \((A_i)_{i \in I}\) be a family of quasi-periodic sets with the same period \(r\), then \(A = \bigcap_{i \in I} A_i\) is quasi-periodic of period \(r\).

Let \(p\) be a fixed prime number, then for every rational number \(q \neq 0\), there exists a unique integer \(u \in \mathbb{Z}\) such that 
\[ q = pu^n\frac{a}{b} \]
where \(a, b \in \mathbb{Z}\) and \(p\) does not divide \(a\) or \(b\). We write this number \(v_p(q) = i\), and by convention \(v_p(0) = \infty\). This is called a \(p\)-adic valuation and it satisfies the following properties:

• for all \(q, q' \in \mathbb{Q}\), \(v_p(qq') = v_p(q) + v_p(q')\),
• for all \(q, q' \in \mathbb{Q}\), \(v_p(q + q') \geq \min(v_p(q), v_p(q'))\),
• for all \(n \in \mathbb{N}\), \(v_p(p^n!) \geq \frac{n}{p-1}\).

Given a polynomial \(P(x) = a_0 + a_1 x + \cdots + a_n x^n\) with rational coefficients, we define its valuation to be \(v_p^j(P) = \min\{v_p(a_j), \ldots, v_p(a_n)\}\) for \(j \leq n\), and \(v_p^j(P) = \infty\) if \(j > n\). It then follows that

• for all \(n \in \mathbb{N}\), \(v_p(P(n)) \geq v_p^0(P)\).

Lemma 72. Let \(P\) be a polynomial with rational coefficients and \(n \in \mathbb{Z}\). Let \(R(x) = (x - m)P(x)\), then for every \(i\), 
\[ v_p^i(P) \geq v_p^{i+1}(R) \]

Proof. Write \(P(x) = a_0 + a_1 x + \cdots + a_n x^n\) and \(R(x) = b_0 + b_1 x + \cdots + b_{n+1} x^{n+1}\), with \(a_n, b_{n+1} \neq 0\). By definition of \(R\), we get that \(b_{n+1} = a_n, b_{i+1} = a_i - ma_{i+1}\) and \(b_0 = -ma_0\). It follows that
\[ a_i = b_{i+1} + mb_{i+2} + \cdots + m^{n-i} b_{n+1}. \]

But then \(v_p(a_i) \geq \min(v_p(b_{i+1}), \ldots, v_p(b_{n+1})) = v_p^{i+1}(P)\) for all \(i\). This implies that \(v_p^i(R) \geq v_p^{i+1}(P)\). \(\Box\)

Proposition 73. Let \((d_n)\) be a sequence of integers and let \(b_n = \sum_{i=0}^{n} \binom{n}{i} p^i d_i\). Then either \(b_n\) is identically 0, or \(\{n : b_n = 0\}\) is finite.

Proof. Assume that \(\{n : b_n = 0\}\) is infinite, we will show that \(b_n\) is identically zero. It is enough to show for all \(n, u \in \mathbb{N}\) that \(v_p(b_n) \geq u\). For any \(n \in \mathbb{N}\), let 
\[ R_n(x) = \sum_{i=0}^{n} \frac{d_i p^i}{i!} x(x-1) \cdots (x-i+1). \]

It follows that for all \(n \geq m\), \(R_m(m) = b_m\). Furthermore, \(v_p^j(R_n) \geq i - \frac{i}{p-1}\) for all \(i \in \mathbb{N}\). Indeed, if \(R_n(x) = \sum_{i=0}^{n} a_i^{(n)} x^i\), then \(a_i^{(n)}\) is a linear combination of \(\frac{d_j p^j}{j!}\) for \(j \geq i\). But 
\[ v_p^j \left( \frac{d_j p^j}{j!} \right) = v_p \left( d_j p^j \right) - v_p \left( p^j \right) \geq \frac{n}{p-1} \geq j - \frac{j}{p-1} \]
and thus \( v_p(a^{(n)}) \geq i - \frac{i}{p-1} \) for all \( i \).

Now fix \( n, u \in \mathbb{N} \), and let \( i \) such that \( i - \frac{i}{p-1} \geq u \). Let \( m_1, \ldots, m_i \) be distincts elements such that \( b_{m_j} = 0 \), and let \( n_0 \geq \max(n, m_1, \ldots, m_i) \). Then \( R_{n_0}(m_j) = b_{m_j} = 0 \) for all \( j \) as we have seen before (since \( n_0 \leq m_j \)). It follows that \( R_{n_0}(x) = (x-m_1) \cdots (x-m_i)P(x) \) for some polynomial \( P \). Thus

\[
v_p(b_n) = v_p(R_{n_0}(q)) = v_p(P(q)) \geq v_p(P) = v_i(P) = i - \frac{i}{p-1} \geq u
\]

by assumption on \( i \).

\[ \Box \]

**Proposition 74.** Let \( \langle S, A, T \rangle \) be LDS with integer coefficients and \( A \) invertible. If \( p > 2 \) does not divide \( \det A \), then \( \{ n : SA^nT = 0 \} \) is quasi-periodic of period \( r < p^2 \).

**Proof.** For any \( n \in \mathbb{N} \), let \( \bar{n} \) denote the class of \( n \) modulo \( p \) and extend it to \( \bar{A} \) coefficient-wise. Then \( \bar{A} \) is a matrix with coefficients in the field \( \mathbb{F} = \mathbb{Z}/p\mathbb{Z} \), but since \( p \) does not divide \( \det A \), then \( \bar{A} \) is invertible over \( \mathbb{F} \). It follows (by Lagrange’s theorem applied to \( GL_k(\mathbb{F}) \)) that there exists \( r < p^2 \) such that \( A^r = I \) and thus \( A^r = I + pM \) where \( M \) is an integer coefficient matrix.

Let \( j \in \{0, \ldots, r-1 \} \) and for all \( n \in \mathbb{N} \), let \( d_n = (SA^j)M^nT \), then

\[
\{ n : u_{j+r} = 0 \} \text{ is either finite or everything. Since there are finitely many } j, \text{ then } \{ n : u_n = 0 \} \text{ is quasi-periodic.} \]

\[ \Box \]

**Theorem 75 (Skolem–Mahler–Lech).** Let \((u_n)_n\) be a LRS, then the set \( Z = \{ n : u_n = 0 \} \) is a ultimately-periodic.

**Proof.** We will show this result in the case of rational LRS only, and admit the general case. By Proposition 54, there exists a LDS \( \langle S, A, T \rangle \) such that \( u_n = SA^nT \). Since it is rational, there exists \( m \in \mathbb{Z} \) such that \( \langle mS, mA, mT \rangle \) is an integer LDS and clearly, \( SA^nT = 0 \) if and only if \( (mS)(mA^n)(mT) = m^{n+2}SA^nT = 0 \). Thus we can assume that \( \langle S, A, T \rangle \) has integer coefficients. Let \( d \) be the dimension of \( A \) and \( V = A^d(\mathbb{R}^d) \), then observe that \( A \) is invertible over \( \mathbb{V} \). Furthermore,

\[
\{ n : u_n = 0 \} = \{ n : SA^nT = 0 \} = \{ n : d_{n+2} : SA^nT = 0 \} \cup \{ d_n : SA^nT = 0 \}.
\]

The first part is finite and the second part corresponds to the LDS \( \langle S, A, T \rangle \). Since \( A \) is invertible over \( \mathbb{V} \) and \( AV \subseteq V \), we can find another LDS \( \langle S', B, T' \rangle \) such that \( SA^nT = S'B^nT \) and \( B \) is invertible (see Exercise 76). Then apply Proposition 74 to \( \langle S', B, T' \rangle \) to conclude. Note that the resulting set is only ultimately periodic and not quasi-periodic, because of the shift \( d_n \) introduced to make \( A \) invertible.

\[ \Box \]

**Exercise 76.** Let \( \langle S, A, T \rangle \) be a LDS and \( V \) a linear subspace. Assume that \( T \in V, AV \subseteq V \) and \( A \) is invertible over \( V \). Show that there exists a LDS \( \langle S', B, T' \rangle \) such that \( SA^nT = S'B^nT' \) and \( B \) is invertible.

**References**


A Exercises

Exercise 77. Consider the following language over alphabet $A = \{a, b\}$:

$$L = \{a^{n_1}ba^{n_2}b \cdots a^{n_k}ba^* : k > 1, \exists i > 1, n_1 = n_i \}.$$

(a) Show that $L$ is a context-free language. If you don’t know context-free languages, you can ignore the question.

We now assume that $L = L_A(\lambda)$ for some probabilistic automaton $A = \langle A, Q, S, \mu, T \rangle$. The goal of this exercise is to reach a contradiction, therefore showing that $L$ is not stochastic. Let $P(x) = c_0 + c_1x + \cdots + c_dx^n$ be the characteristic polynomial of $\mu(a)$. Recall that by Theorem 53, $P(\mu(a)) = 0$.

(b) Recall why 1 is an eigenvalue of $\mu(a)$. Show that $c_0 + \cdots + c_d = 0$ and that for any word $w$, $\sum_{i=0}^d c_i A(a^i w) = 0$.

Let $\text{Pos} := \{ i : c_i > 0 \}$ and $\text{NonPos} := \{ i : c_i \leq 0 \}$. Define $w = ba^{i_1}b \cdots ba^{i_k}b$ where $\{i_1, \ldots, i_k\} = \text{Pos}$.

(c) Let $i \in \{0, \ldots, n\}$, when is $a^i w \in L$? Show that $\sum_{i=0}^d c_i A(a^i w) > \sum_{i=0}^d c_i \lambda$. Why is this a contradiction?

Exercise 78. Consider the following language over alphabet $A = \{a, b, c\}$:

$$L = \{a^n b^m c^n : n > 0 \}.$$

It is a classical result that $L$ is a not a context-free language. The goal of this exercise is to show that $L$ is stochastic.

(a) Show that $L = L_1 \cap L_2$ where $L_1 = \{a^n b^m c^n : n > 0 \}$ and $L_2 = \{a^n b^m c^n : n > 0 \}$ where $x^+ := xx^*$.

(b) Build an automaton $A_1$ such that $A_1(b^m c) = 2^{-m}$ if $m > 0$ and 0 otherwise. Then modify it into $B_1$ such that $B_1(a^* b^m c^+ ) = 2^{-m}$ if $m > 0$ and 0 otherwise.

(c) Build an automaton $A_2$ such that $A_2(a^n b) = 1 - 2^{-n}$ if $n > 0$ and 0 otherwise. Then modify it into $B_2$ such that $B_2(a^n b^+ c^+ ) = 1 - 2^{-n}$ if $n > 0$ and 0 otherwise.

(d) Build an automaton $C_1$ such that $C_1(a^n b^m c^+ ) = \frac{1}{2}(1 - 2^{-n} + 2^{-m})$ if $n, m > 0$ and 0 otherwise.

(e) Show that $L_1 = L_1^C(\frac{1}{2})$.

(f) Show that for all $x, y \in [0, 1]$, $x = \frac{1}{2}$ and $y = \frac{1}{2}$ if and only if $\frac{1}{2}x(1-x) + \frac{1}{2}y(1-y) = \frac{1}{4}$.

(g) Show that for any two automata $A$ and $B$, there exists an automaton $C$ such that $L_C^C(\frac{1}{4}) = L_A^C(\frac{1}{2}) \cap L_B^C(\frac{1}{2})$.

(h) Conclude.

Exercise 79. We will now consider various operations on stochastic languages.

(a) Show that if $L$ is regular then there exists an automaton $A$ such that $A(w) = 1$ if $w \in L$ and 0 if $w \notin L$.

(b) Let $L$ be a regular language, $A$ be a probabilistic automaton and $\lambda$ a threshold, show that there exists $B$ and $\mu, \delta$ such that $L_B(\mu) = L_A(\lambda) \cap L$ and $L_B(\delta) = L_A(\lambda) \cup L$.

(c) Show that if $L = L_A(\frac{1}{2})$ for some automaton $A$ then $L$ is stochastic.

Consider $L = \{a^{n_1}b \cdots ba^{n_k}b : k > 1 \land n_1 = n_k \}$. We will show that $LL'$ is not stochastic for some regular language $L'$.

(d) Build an automaton $A$ such that $A(a^{n_1}b \cdots ba^{n_k}b) = 1 - 2^{1-k-n_1}$ if $k \geq 1$.

(e) Build an automaton $B$ such that $B(a^{n_1}b \cdots ba^{n_k}b) = 2^{1-k-n_1}$ if $k > 1$.

(f) Show that $L$ is a stochastic language.

(g) Show that $LA^* = \{a^{n_1}ba^{n_2}b \cdots a^{n_k}ba^* : k > 1, \exists i > 1, n_1 = n_i \}$ where $A = \{a, b\}$.

(h) Conclude using Exercise 77.

(i) Show that $LcA^*$ is stochastic. Find a homomorphism $h : \{a, b, c\} \rightarrow A^*$ such that $h(LcA^*)$ is not stochastic.

(j) (not easy) Using a similar technique as in Exercise 77, show that $L^*$ is not stochastic by consider the word $w = ba^{i_1}b(a^i b)^2 \cdots (a^i b)^2$.

---

6Also known as algebraic languages, see https://fr.wikipedia.org/wiki/Langage_alg%C3%A9brique.

7In other words, any word not of the form $b^n c$ must have probability of acceptance 0.
Exercise 80. Let $\mathcal{A}$ and $\mathcal{B}$ be two probabilistic automata. We say that they are equivalent if for every word $w$, $\mathcal{A}(w) = \mathcal{B}(w)$.

(a) Write $\mathcal{A} = \langle A, Q_1, S_1, \mu_1, T_1 \rangle$ and $\mathcal{B} = \langle A, Q_2, S_2, \mu_2, T_2 \rangle$. Recall the construction of $\mathcal{C} = \langle A, Q, S, \mu, T \rangle$ such that $C(w) = \frac{1}{2}A(w) + \frac{1}{2}B(w)$. Find a vector $\bar{T}$ such that for every word $w$, $\mathcal{A}(w) = \mathcal{B}(w)$ if and only if $S\mu(w)\bar{T} = 0$.

(b) Define $V_n = \text{span}\{S\mu(w) : |w| \leq n\}$ for all $n \in \mathbb{N}$. Show that if $V_n = V_{n+1}$ for some $n$, then $V_{n+1} = V_{n+2}$.

(c) Define $V = \text{span}\{S\mu(w) : w \in A^*\}$, show that $V = V_{d+d'}$ where $d$ (resp. $d'$) is the number of states of $\mathcal{A}$ (resp. $\mathcal{B}$).

(d) Show that $\mathcal{A}$ and $\mathcal{B}$ are equivalent if and only if $v\bar{T} = 0$ for all $v \in V$.

(e) Show that if $\mathcal{A}$ and $\mathcal{B}$ are not equivalent then there exists $w$ of length at most $d + d'$ such that $\mathcal{A}(w) \neq \mathcal{B}(w)$.

(f) Show that the equivalence problem is in $\text{coNP}$. 
B Solutions to exercises

Exercise 2. We check that $1 + 0 + 0 = 1$ for $I$. Then each line of $\mu(a)$ and $\mu(b)$, for example $0 + \frac{1}{2} + \frac{1}{2} = 1$ and $0 + \frac{1}{2} + \frac{3}{4}$. 

Exercise 3. Let $M \in [0,1]^{P \times Q}$ and $N \in [0,1]^{Q \times R}$ then $(MN)_{p,r} = \sum_{q \in Q} M_{p,q} N_{q,r}$. It follows that on line $p$ we have

$$\sum_{r \in R} (MN)_{p,r} = \sum_{r \in R} \sum_{q \in Q} M_{p,q} N_{q,r} = \sum_{q \in Q} M_{p,q} \sum_{r \in R} N_{q,r} = \sum_{M} p,q = 1.$$ 

Exercise 4. Intuitively, $\mu(w)_{q,q'}$ is the probability that we end up in state $q'$ by reading word $w$ from state $q$. Formally, $\mu(w)_{q,q'}$ is the sum of the weights (probabilities) of all paths from $q$ to $q'$ that are labelled by $w$. Indeed, this is true when $w$ is just one letter, by definition. Let $w \in \Sigma^*$ and $a \in \Sigma$ then any path $q \xrightarrow{w} q'$ is of the form $q \xrightarrow{w[y]} q'' \xrightarrow{a[z]} q$ where $q'' \in A$, $z = (a)_{q'',a}$ and $x = yz$. Summing over all such paths with fixed $q''$ gives a probability of $\mu(w)_{q,q''} \mu(a)_{q'',q'}$ by induction. Therefore the sum of all paths from $q$ to $q'$ labelled by $w$ is

$$\sum_{q'' \in Q} \mu(w)_{q,q''} \mu(a)_{q'',q'} = (\mu(w) \mu(a))_{q,q'} = \mu(wa)_{q,q'}.$$ 

Then $S_\mu(w)$ is the probability distribution of the states starting from the initial distribution $I$. This is indeed a distribution because it is a stochastic vector.

Exercise 5. In the first approach, we simply write $\mathcal{B}$ using a substochastic matrix: $\mathcal{B} = \langle A, Q, S, \mu, T \rangle$ where $A = \{a, b\}$, $Q = \{p, q, r\}$ and

$$S = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \mu(a) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix}, \quad \mu(b) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$ 

In the second approach, we create a sink state $\perp$ to account for the missing probability: $\mathcal{B}' = \langle A, Q', S', \mu', T' \rangle$ where $Q' = \{q, r, \perp\}$ and

$$S' = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \mu'(a) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix}, \quad \mu'(b) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad T' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$ 

Notice that indeed $\mu'(a)$ and $\mu'(b)$ are stochastic. Furthermore, we have the following relationship between the two automata, for every letter $x \in A$, vector $v \in \mathbb{Q}^3$ and “sink probability” $\varepsilon \in \mathbb{Q}$:

$$\mu'(x) \begin{bmatrix} v \\ \varepsilon \end{bmatrix} = \begin{bmatrix} \mu(x)v \\ \mu'(x)\varepsilon \end{bmatrix}$$

for some $\varepsilon'$. Thus for every word $w$,

$$S' \mu'(w) T = \begin{bmatrix} I & 0 \end{bmatrix} \mu'(w) \begin{bmatrix} T \\ 0 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \mu(w)T \\ \varepsilon \end{bmatrix} = S \mu(w) T.$$ 

Exercise 7. Check that $\mathcal{A}(bba) = \frac{1}{2}$ and $\mathcal{A}(abb) = \frac{2}{3}$. Thus $bba \not\in \mathcal{L}(\frac{1}{2})$ and $abb \in \mathcal{L}(\lambda)$ for all $\lambda < \frac{2}{3}$. In fact $\mathcal{L}(\frac{2}{3}) = \emptyset$ as we will see. Check that $\mathcal{A}((ab)^n b) = \frac{2}{3}$ for every $n \in \mathbb{N}$, thus $(ab)^* b \subseteq \mathcal{L}(\lambda)$ for all $\lambda < \frac{2}{3}$.

Exercise 8. The edges of $\mathcal{B}$ are exactly the edges of $\mathcal{A}$ labelled by $a$, thus $\mathcal{L}_{\mathcal{B}}(\lambda) = \mathcal{L}(\lambda) \cap a^*$. 

Exercise 9. TODO

Exercise 10. Each regular language can be described by a regular expression, that is a finite word over the finite alphabet $A \cup \{(,), +, *, \varepsilon\}$. The set of words over a finite alphabet is countable.

Exercise 13. One immediately checks that $\equiv_L$ is reflexive, symmetric and transitive. Let $L$ be a regular language and let $\mathcal{A} = \langle A, Q, q_0, \delta, q_f \rangle$ be a deterministic finite automaton, where $q_0, q_f$ are the initial and final states and $\delta : Q \times A \rightarrow Q$ is the transition function (which we can assume is total), which we naturally extend to words in the obvious way. For each state $q$, define $L_q = \{ w \in A^* : \delta(q_0, w) = q \}$ to be the set of words $w$ such that the automaton is in state $q$ after reading $w$ from $q_0$. We claim that for all $u, v \in L_q$, $u \equiv_L v$. Indeed, if $u \in L_q$ and $v \in A^*$, then $\delta(q_0, uw) = \delta(\delta(q_0, u), w) = \delta(q, w)$ thus $uw \in L$ if and only if $\delta(q, w) = q_f$. Note that this condition is independent of $u \in L_q$ and thus $uw \in L$ if and only if $vw \in L$. 

28
This probabilistic automaton is represented by the tuple \((A, Q, \mu, T)\) where \(A = \{a, b\}\), \(Q = \{p, q\}\) and 
\[
S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mu(a) = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}, \quad \mu(b) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
\]

One checks that 
\[
\mu(a)^n \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2^{-n}x + (1 - 2^{-n})y \\ y \end{bmatrix}, \quad \mu(b) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ x \end{bmatrix}, \quad \mu(a)^n \mu(b) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (1 - 2^{-n})x \\ x \end{bmatrix}.
\]

Therefore, 
\[
S \mu(x(n_1, \ldots, n_k)) T = S \mu(a)^{n_1} \mu(b) \cdots \mu(a)^{n_k} \mu(b) T = S \mu(a)^{n_1} \mu(b) \cdots \mu(a)^{n_k} \mu(b) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = (1 - 2^{-n_k})
\]

Let \(u = x(n_1, \ldots, n_k)\) and \(w = x(n_{k+1}, \ldots, n_l)\) then \(uw = x(n_1, \ldots, n_l)\) thus \(C(uw) = C(u)C(w)\) by a straightforward calculation. To see the density, fix \(\lambda \in (0, 1)\) and let \(\mu_{\infty} = \log \lambda < 0\). Now consider the sequence defined by \(\mu_0 = 0\) and \(\mu_{i+1} = \mu_i + \log(1 - 2^{-n_i})\) where \(n_{i+1} = \min\{k \geq 1 : \mu_i + \log(1 - 2^{-k}) > \mu_{\infty}\}\). Such a \(k\) exists because \(\mu_i > \nu_{\infty}\) and \(\log(1 - 2^{-k}) \to 0\) as \(k \to \infty\). Then \(\mu_k \to \mu_{\infty}\) as \(k \to \infty\) thus \(e^{\mu_k} \to \lambda\) as \(k \to \infty\). But \(e^{\mu_k} = \prod_{i=1}^{k} (1 - 2^{-n_i}) = C(x(n_1, \ldots, n_k))\). The proof that \(C\) is universally non-regular is then the same as for Theorem 15.

**Exercise 19.** Since \(L\) is nonempty, there exists \(x \in L\), which must therefore have length \(|x| \geq n\). For every \(1 \leq i \leq n\), let \(u_i = x_1 \cdots x_i\). Then \(u_i \not\in L\), \(u_i \not\in L\) for \(i < j\). Indeed, if we let \(w = x_{j+1} \cdots x_n\) then \(u_j w = x \in L\) but \(|u_i| = i + n - j < n\) \(u_i w \not\in L\). It follows that \(x \in L\) has at least \(n\) equivalence classes. By Theorem 12, any deterministic finite automaton that recognizes \(L\) must therefore have at least \(n\) states.

**Exercise 30.** Let \(A = (A, Q, S, \mu, T)\) be a probabilistic automaton and let \(p\) be the smallest integer such that for all \(a \in A\), \(2^p \mu(a)\) has integer entries. In other words, \(p\) is the highest power of 2 appearing in the denominators of the transition probabilities. If \(p = 0\) or \(p = 1\) then \(A\) is simple already. We now give the intuition: for each state \(q\) and letter \(a\), we will build a tree of height \(p\) such that each leaf has probability \(2^{-p}\) to be reached from \(q\) after reading \(a^p\). But since \(p\) is such that \(2^p \mu(a)\) has integer entries, it means that we simply need to choose \((2^p \mu(a))_{q,a}\) leaves for each \(a\) and put a transition with probability 1 from this leaf to \(p\). Graphically, for example,

\[
\begin{align*}
\text{Exercise 32.} & \quad \text{If } A \text{ and } B \text{ are simple then it is clear that all probabilities that appear in } C \text{ are product of the form } xy \text{ where } x \text{ and } y \text{ are multiple of } \frac{1}{2}, \text{ therefore they are multiple of } \frac{1}{4}.
\end{align*}
\]

**Exercise 34.** TODO

**Exercise 38.** TODO
Exercise 77. It is clear that any such path from 1 to 3 must go through the cycle 1 \to 2 \to 1 at most \( k - 1 \) times, then follow 1 \to 3 once and then 3 \to 3 for the remaining time. Thus
\[
A_x \left( 1 \xrightarrow{w} 3 \right) = \sum_{i=0}^{k-1} A_x \left( 1 \xrightarrow{a^n b \cdots a^n b} 1 \right) A_x \left( 1 \xrightarrow{a^{n+1} b} 3 \right) A_x \left( 3 \xrightarrow{a^{n+2} b \cdots a^{n+2} b} 3 \right)
\]
and similarly
\[
A_x \left( 4 \xrightarrow{w} 6 \right) = \sum_{i=0}^{k-1} A_x \left( 4 \xrightarrow{a^n b \cdots a^n b} 4 \right) A_x \left( 4 \xrightarrow{a^{n+1} b} 6 \right) A_x \left( 6 \xrightarrow{a^{n+2} b \cdots a^{n+2} b} 6 \right)
\]
Exercise 56. If we follow the proof of the course then we need find \( a, b \in \mathbb{Q} \) such that
\[
A^2 = aA + ba^0 \Leftrightarrow \left[ \begin{array}{cc} -1 & 2 \\ -2 & 3 \end{array} \right] = \left[ \begin{array}{cc} b & a \\ -a & 2a + b \end{array} \right] \Leftrightarrow a = 2 \land b = -1.
\]
Therefore we get that \( u_{n+2} = 2u_{n+1} - u_n, \quad u_0 = SA^0T = 0 \) and \( u_1 = SA^1T = 1 \). It is not hard to see that \( u_n = n \). It is also immediate that
\[
A^n \left[ \begin{array}{c} u_n \\ u_{n+1} \end{array} \right] = \left[ \begin{array}{c} u_{n+1} \\ 2u_{n+1} - u_n \end{array} \right] = \left[ \begin{array}{c} u_{n+1} \\ u_{n+2} \end{array} \right].
\]
One easily checks by induction that for every \( n \in \mathbb{N} \),
\[
B^n = \left[ \begin{array}{cc} 1 & n \\ 0 & 1 \end{array} \right]
\]
and thus \( SB^nT = n = u_n \), but
\[
B^n \left[ \begin{array}{c} u_n \\ u_{n+1} \end{array} \right] = \left[ \begin{array}{c} u_n + u_{n+1} \\ u_{n+1} \end{array} \right] \neq \left[ \begin{array}{c} u_{n+1} \\ u_{n+2} \end{array} \right].
\]
Exercise 77. (a) We show \( L \) can be recognized by a nondeterministic automaton with one counter. The automaton first counts to \( n_1 \) until it reaches the first \( b \). It then guesses the occurrence of a \( b \) and starts decreasing the counter for every \( a \) until the next \( b \). If the counter reaches 0, the word is accepted.

It is also possible to write a grammar for this language:
\[
S \to RB \quad R \to aRa \mid Bb \mid b \quad B \to Ba \mid Bb \mid b.
\]
Indeed let \( A = \{ a, b \} \) then \( L(B) = ba^* \) thus \( L(R) = \bigcup_{n>0} a^n(L(B)b + b)a^n = \bigcup_{n>0} a^n(ba^*b + b)a^n \) and therefore
\[
L(S) = \bigcup_{n>0} a^n(ba^*b + b)a^nA^*.
\]
(b) Since \( \mu(a) \) is stochastic, we get that
\[
\mu(a) = \left[ \begin{array}{c} \mu(a)_{1,1} + \cdots + \mu(a)_{1,d} \\ \vdots \\ \mu(a)_{d,1} + \cdots + \mu(a)_{d,d} \end{array} \right] = \left[ \begin{array}{c} 1 \\ \vdots \\ 1 \end{array} \right]
\]
which shows that 1 is an eigenvalue. It follows that \( P(1) = 0 \) since \( P \) is the characteristic polynomial and thus \( P(\mu(a)) = 0 \) by Theorem 53. Unfolding the definition, we get that \( c_0 \lambda_d + c_1 \mu(a) + \ldots + c_d \mu(a)^d = 0 \). Using that \( \mu \) is a morphism, we then have that

\[
\sum_{i=0}^{d} c_i A(a^i w) = \sum_{i=0}^{d} c_i S \mu(a)^i \mu(w) T = S \left( \sum_{i=0}^{d} c_i \mu(a)^i \right) \mu(w) T = 0.
\]

(c) By definition of \( L \), \( a^i w \in L \) if and only if \( i \in \text{Pos} \). Then

\[
\sum_{i=0}^{d} c_i A(a^i w) = \sum_{i=\in Pos} c_i A(a^i w) + \sum_{i=\notin \text{Pos}} c_i A(a^i w)
\]

\[
> \sum_{i=\in Pos} c_i \lambda + \sum_{i=\notin \text{Pos}} c_i A(a^i w) \quad \text{since } a^i w \in L \text{ thus } A(a^i w) > \lambda
\]

\[
> \sum_{i=\in Pos} c_i \lambda + \sum_{i=\notin \text{Pos}} c_i \lambda \quad \text{since } a^i w \notin L \text{ thus } A(a^i w) \leq \lambda \text{ and } c_i \leq 0
\]

\[
= \sum_{i=0}^{d} c_i \lambda = 0.
\]

This is a contradiction because we have seen that \( \sum_{i=0}^{d} c_i A(a^i w) = 0 \) and \( \sum_{i=0}^{d} c_i \lambda = \left( \sum_{i=0}^{d} c_i \right) \lambda = 0 \) thus \( 0 > 0 \).

**Exercise 78.** (a) Clearly \( L_1 \cap L_2 \subseteq a^+ b^+ c^+ \). Then \( a^n b^n c^\omega \in L_1 \cap L_2 \) if and only if \( n = m \) (by virtue of being in \( L_1 \)) and \( m = p \) (by virtue of being in \( L_2 \)). Thus \( L = L_1 \cap L_2 \).

(b) Consider the automaton \( A_1 \) below, it is substochastic only but can trivially be made stochastic.

![Diagram](image)

It is clear that any word not in the language \( b^+ c \) has probability of acceptance 0. Furthermore, a direct computation shows that \( A_1(b^\omega c) = 2^{-m} \) if \( m > 0 \). We then add a self-loop to \( p \) to accept any \( a \) and a self-loop to \( f \) to accept any \( c \) to obtain \( B_1 \).

(c) Consider the automaton \( A_2 \) below, again it is substochastic only.

![Diagram](image)

Check that any word not in \( a^+ b \) has probability of acceptance 0. Then observe that \( A_2 \left( p \xrightarrow{a^n} p \right) = 2^{-n} \) and by substochasticity \( A_2 \left( p \xrightarrow{a^n} x \right) = 1 - A_2 \left( p \xrightarrow{a^n} p \right) = 1 - 2^{-n} \) since after reading \( a^n \), the automaton is either state \( p \) or \( x \). Finally, only state \( x \) leads to \( f \) when reading \( b \) therefore \( A_2(a^n b) = A_2 \left( p \xrightarrow{a^n} x \right) = 1 - 2^{-n} \).

Another clever solution found by a student is the following: take \( A_1 \), replace \( b \) by \( a \) and \( b \) by \( c \) and take the complement. This automaton satisfies \( A_2(a^n b) = 1 - 2^{-n} \) but has probability of acceptance 1 for the other words. But notice that \( a^+ b^+ \) is regular so we can build \( C \) such that \( C(w) = 1 \) if \( w \in a^+ b^+ \) and 0 otherwise. Then the product of the two automata gives the result.

We then obtain \( B_2 \) by making \( f \) non-final, adding a state \( f' \) with an arrow from \( f \) to \( f' \) labelled by \( c \), adding a self-loop to \( f \) to accept any \( b \) and a self-loop to \( f' \) to accept any \( c \).

(d) Build \( C_1 \) such that \( C_1(w) = \frac{1}{2} B_1(w) + \frac{1}{2} B_2(w) \), then

\[
C_1(a^n b^m c^+) = \frac{1}{2} B_1(a^{n-1} b^m c^+) + \frac{1}{2} B_2(a^{n-1} b^m c^+) = \frac{1}{2} 2^{-m} + \frac{1}{2} (1 - 2^{-n}) = \frac{1}{2} (1 + 2^{-m} - 2^{-n}).
\]
(e) Any word not in \( a^+b^+c^+ \) has probably of acceptance 0, and by the previous computation, \( C_1(a^nb^mc^+) = \frac{1}{2} \) if and only if \( n = m \). Therefore \( \mathcal{L}_{C_1}^\pi(\frac{1}{2}) = \{a^nb^mc^+ : n = m \} = L_1 \).

(f) Clearly if \( x = y = \frac{1}{2} \) then \( \frac{1}{2}x(1 - x) + \frac{1}{2}y(1 - y) = \frac{1}{4} \). Conversely, observe that \( x(1 - x) \geq \frac{1}{4} \) for all \( x \), and similarly for \( y \). Thus if \( \frac{1}{2}x(1 - x) + \frac{1}{2}y(1 - y) \neq \frac{1}{4} \) then either \( x(1 - x) > \frac{1}{4} \) or \( y(1 - y) > \frac{1}{4} \) and thus either \( x \neq \frac{1}{2} \) or \( y \neq \frac{1}{2} \).

(g) Without loss of generality we can assume that both automata have the same alphabet by taking the intersection (since any word in the resulting intersection must be in both). Write \( A = \langle A, Q_1, S_1, \mu_1, T_1 \rangle \) and \( B = \langle A, Q_2, S_2, \mu_2, T_2 \rangle \), define \( C = \langle A, Q', S', \mu', T' \rangle \) where

\[
S = \begin{bmatrix} S_1 & S_2 \end{bmatrix}, \quad \mu(a) = \begin{bmatrix} \mu_1(a) & 0 \\ 0 & \mu_2(a) \end{bmatrix}, \quad T = \begin{bmatrix} T_1 & T_2 \end{bmatrix}.
\]

Note that \( S \) is indeed stochastic. Then one easily checks that \( \mathcal{C}(w) = S\mu(w)T = \frac{1}{2}(S_1\mu_1(w)T_1 + S_2\mu_2(w)T_2) = \frac{1}{2}(\mathcal{A}(w) + \mathcal{B}(w)) \). Using the observation of (f), we get the result.

(h) Similarly to \( C_1 \), we can build \( C_2 \) such that \( \mathcal{L}_{C_2}^\pi(\frac{1}{2}) = L_2 \). Then there exists \( C \) such that \( \mathcal{L}_{C}^\pi(\frac{1}{2}) = \mathcal{L}_{C_1}^\pi(\frac{1}{2}) \cap \mathcal{L}_{C_2}^\pi(\frac{1}{2}) = L_1 \cap L_2 = L \).

**Exercise 79.**

(a) Trivial.

(b) Let \( B \) be as above for \( L \), then build automaton \( C \) such that \( \mathcal{C}(w) = \frac{1}{2}\mathcal{A}(w) + \frac{1}{2}\mathcal{B}(w) \). Using that \( \mathcal{B}(w) \in \{0, 1\} \) we get that

\[
\mathcal{C}(w) > \frac{1}{2} \iff \mathcal{B}(w) = \frac{1}{2} \lor \mathcal{A}(w) > \lambda \iff w \in L \lor w \in \mathcal{L}_A(\lambda).
\]

Similarly,

\[
\mathcal{C}(w) > \frac{1 + \lambda}{2} \iff \mathcal{B}(w) = \frac{1}{2} \land \mathcal{A}(w) > \lambda \iff w \in L \land w \in \mathcal{L}_A(\lambda).
\]

(c) Use that \( x(1 - x) \geq \frac{1}{4} \) if and only if \( x = \frac{1}{2} \).

(d) Check that the automaton below is stochastic.

Check that \( \mathcal{A}(a^{n_1}b\cdots ba^{n_k}b) = 1 - 2^{1-k-n_1} \) if \( k \geq 1 \). Indeed, after reading \( a^{n_1}b\cdots ba^{n_k}b \) with \( k \geq 1 \), the automaton can only be in \( f \) or \( x \). But since it is stochastic, the probability of acceptance (which is the probability of being in \( f \)), is 1 minus the probability of being in \( x \), which is \( 2^{1-k-n_1} \). Any other word has probability of acceptance 0.

(e) Check that the following automaton satisfies \( \mathcal{B}(a^{n_1}b\cdots ba^{n_k}b) = 2^{1-k-n_1} \) if \( k > 1 \).

(f) Build \( C \) such that \( \mathcal{C}(w) = \frac{1}{2}(\mathcal{A}(w) + \mathcal{B}(w)) \). Then \( \mathcal{C}(a^{n_1}b\cdots ba^{n_k}b) = \frac{1}{2}(1 + 2^{1-k-n_1} - 2^{1-k-n_1} \mathcal{A}(w)) \). It follows that \( \mathcal{L}_{C}^\pi(\frac{1}{2}) = L \) and thus \( L \) is stochastic.

(g) Trivial.

(h) \( LA^* \) is not stochastic, thus concatenation of a stochastic and a regular language is not necessarily stochastic.

(i) Clearly \( cA^* \) is regular and since \( c \) is not in the alphabet of \( L \), \( LcA^* \) is stochastic: the letter \( c \) acts as a reset to go from one automaton to another. The morphism \( h(a) = a, h(b) = b, h(c) = \varepsilon \) is such that \( h(LcA^*) = LA^* \) which is not stochastic.
(j) Assume that $L^* = \mathcal{L}_A(\lambda)$ for some probabilistic automaton $A = (A, Q, S, \mu, T)$. Let $P(x) = c_0 + c_1 x + \ldots + c_d x^n$ be the characteristic polynomial of $\mu(a)$. Recall that by Theorem 53, $P(\mu(a)) = 0$. Since 1 is an eigenvalue of $\mu(a)$, $c_0 + \ldots + c_d = 0$ and for any word $w$, $\sum_{i=0}^d c_i A(a^i w) = 0$. Define $w = b a^i b (a^i b)^2 \ldots (a^i b)^2$ where $\{i_1, \ldots, i_k\} = \{i : c_i > 0\}$, then $a^i w \in L$ if and only if $i \in \{i_1, \ldots, i_k\}$ but similarly reasoning to Exercise 77 shows that $\sum_{i=0}^d c_i A(a^i w) > \sum_{i=0}^d c_i \lambda$ which is absurd.

Exercise 80.

(a) Define $C = (A, Q', S', \mu', T')$ where

$$S = \frac{1}{2} \begin{bmatrix} S_1 & S_2 \end{bmatrix}, \quad \mu(a) = \begin{bmatrix} \mu_1(a) & 0 \\ 0 & \mu_2(a) \end{bmatrix}, \quad T = \begin{bmatrix} T_1 & T_2 \end{bmatrix}, \quad \tilde{T} = \begin{bmatrix} T_1 & -T_2 \end{bmatrix},$$

Note that $S$ is indeed stochastic. Then check that

$$C(w) = S \mu(w) T = \frac{1}{2} (S_1 \mu_1(w) T_1 + S_2 \mu_2(w) T_2) = \frac{1}{2} (\mathcal{A}(w) + \mathcal{B}(w)).$$

On the other hand, we have that

$$S \mu(w) \tilde{T} = \frac{1}{2} (S_1 \mu_1(w) T_1 - S_2 \mu_2(w) T_2) = \frac{1}{2} (\mathcal{A}(w) - \mathcal{B}(w))$$

thus $S \mu(w) \tilde{T}$ if and only if $\mathcal{A}(w) = \mathcal{B}(w)$.

(b) Observe\(^8\) that for any word $w$ and letter $a$, $S \mu(wa) = (S \mu(w))\mu(a) \in V_{|w|}\mu(a)$. Thus $V_{n+1} = V_n \cup \cup_{a \in A} V_n \mu(a)$. Therefore if $V_n = V_{n+1}$ for some $n$, then

$$V_{n+2} = V_{n+1} \cup \bigcup_{a \in A} V_{n+1} \mu(a) = V_n \cup \bigcup_{a \in A} V_n \mu(a) = V_{n+1}.$$  

(c) By the previous question, for every $n$, either $V_n = V_{n+1}$ or $\dim V_n < \dim V_{n+1}$. But $V_n \subseteq \mathbb{R}^{d+d'}$ thus $\dim V_n \leq d + d'$. It follows that the sequence $V_n = V_{d+d'}$ for all $n \geq d + d'$. But since $V = \bigcup_{n \in \mathbb{N}} V_n$, we get that $V = V_{d+d'}$.

(d) If $\mathcal{A}$ and $\mathcal{B}$ are equivalent then $S \mu(w) \tilde{T} = 0$ for all $w$. By linearity, and since the $S \mu(w)$ span $V$, it follows that $v \tilde{T} = 0$ for all $v \in V$. Conversely if $\mathcal{A}$ and $\mathcal{B}$ are not equivalent, then there exists $w \in A^*$ such that $S \mu(w) \tilde{T} \neq 0$. But $S \mu(w) \in V$ so $\exists v \in V$ such that $v \tilde{T} \neq 0$.

(e) If $\mathcal{A}$ and $\mathcal{B}$ are not equivalent, then there exists $v \in V$ such that $v \tilde{T} \neq 0$. But $V = V_{d+d'} = \text{span}(S \mu(w) : |w| \leq d+d')$ thus there must exists $w$ with $|w| \leq d+d'$ and $S \mu(w) \tilde{T} \neq 0$ (otherwise by linearity every vector $v \in V$ would satisfy $v \tilde{T} = 0$). Finally $S \mu(w) \tilde{T} \neq 0$ implies that $\mathcal{A}(w) \neq \mathcal{B}(w)$.

(f) To show that the equivalence problem is in \textbf{coNP}, it is equivalent to show that the disequivalence problem (decide whether two automata are not equivalent) is in \textbf{NP}. Thanks to the previous question, this is equivalent to searching a word $w$ of linear size $(d+d')$ such that $\mathcal{A}(w) \neq \mathcal{B}(w)$. This can be done in \textbf{NP} by guessing such a word, computing $\mathcal{A}(w)$ and $\mathcal{B}(w)$ and comparing them. Note that we can compute $\mathcal{A}(w)$ in polynomial time because the numbers are rational and their size remains bounded by a polynomial.

---

\(^8\)If $X$ is a set and $M$ a matrix, $XM := \{xM : x \in X\}$. 