UNIVERSITÀ DEGLI STUDI DI MILANO

SCUOLA DI DOTTORATO IN
Informatica

DIPARTIMENTO DI
Informatica

CORSO DI DOTTORATO
Informatica
XXV° Ciclo

Descriptional complexity of classical and quantum unary automata

INF/01

Dottoranda:
Maria Paola BIANCHI

Relatore:
Prof. Beatrice PALANO
Correlatore:
Prof. Carlo MEREGHETTI
Coordinatore del Dottorato:
Prof. Ernesto DAMIANI

Anno Accademico 2011/12

Acknowledgements

I would like to thank Beatrice Palano and Carlo Mereghetti for supervising my studies since long before I started my phd. Throughout my whole academic career they supported and guided me with extraordinary patience and dedication. Knocking at their door 7 years ago was definitely the best choice I could make.

I would also like to thank all the professors, researchers and phd students with whom I collaborated during my phd. In particular, I owe a huge thanks to Alberto Bertoni and Giovanni Pighizzini, without whom many of the results in this thesis would have never seen the light. The time spent with them made me feel honored to have the chance of working with such brilliant minds.

Contents

Introduction

Automata theory

Formalizing and understanding the concept of computation has always been a fundamental issue in theoretical computer science. The best way for getting a deeper insight into the power and features of a computational device, without depending on the specific architecture of a physical machine, is to consider an abstract model, which can be described and manipulated through a mathematical representation. The first and most important of these abstractions is the Universal Turing Machine (UTM), which is based on a set of principles stated by Alan Turing in 1936 and elaborated later by John von Neumann in the 1940's. This model, composed by a working tape and a table of rules, is very simple yet extremely powerful, as it is able to simulate the logic of any computer algorithm.

Over the following decades, scientists have proposed many other mathematical abstractions for modeling simpler computational devices with limited resources. Among these, finite state automata represent a theoretical model for all algorithms and electronic devices such that, in every computational step, the current configuration (*state*) summarizes completely the whole process computed up to that point (see, e.g., [33]). Hence, for each elementary unit of information (*symbol*) in input, the evolution of the system (*next state*) depends completely on the input symbol and the current state. The input is presented to the device on an infinite tape, accessible through a reading head, which scans the tape with a single movement from left to write (*one-way* model). The output of an automaton is a boolean accept/reject value, therefore the behavior of the device can be summarized by the set of input strings (also called *words*) which lead to a positive outcome. This set of strings is called the *language accepted* by the automaton.

In this thesis we consider three types of one-way finite automata, which differ from each other for the way evolution is defined.

- *Deterministic* automata (DFAs) [50]: they have their starting configuration in an *initial state*, a deterministic *transition* associated to each symbol and a set of *final states*. The language recognized by a DFA is the set of words

that bring the system from the initial state to one of the final states, through the sequence of transitions associated with the symbols in the word.

- *Nondeterministic* automata (NFAs) [50]: they are defined like DFAs, but the same symbol can trigger more than one transition from a given state to a set of possible next states. Therefore, in every moment, the system configuration is described by a set of states instead of a single one. We also allow multiple initial states. The language recognized by a NFA is the set of words for which there exists a possible computation from one of the initial states to one of the final states through the sequence of transitions associated to the symbols in the word.

- *Probabilistic* automata (PFAs) [49]: like NFAs, for a given state, a symbol may determine more than one possible next state. However, in this case, the automaton can perform only one of those transitions, chosen according to a certain probability distribution. The evolution is therefore stochastic and, instead of an initial set of states, we have an initial probability distribution. Each input word has a certain probability of being accepted, and the language accepted by the automaton is the set of words whose acceptance probability exceeds a fixed value called *cut point*. If, for each word, the probability of acceptance cannot be arbitrarily close to the cut point, we call it *isolated*. In this thesis we only consider acceptance with an isolated cut point.

The *computational power* of all these models is the same, as they all recognize regular languages [49, 50], but they may require a different number of states for recognizing the same language: as we show in this thesis, for many families of regular languages, PFAs can be smaller than NFAs, which generally require fewer states than DFAs.

Quantum models

Moore's law is a well-known statement describing performance increasing of computing systems along years [44]. In 1975, on the basis of actual observations on a period of 16 years, Moore emphasizes the growth by a factor of 2 per year of the number of transistors within a chip. This leads him to state such an exponential increase as a law for microelectronics development and, up to now, such a law has demonstrated its validity. Currently, quantum effects start manifesting in the behavior of electronic devices as their size becomes smaller. This implies that such phenomena cannot be ignored in the construction of future computers.

The idea of constructing a computational model able to represent a system subject to the laws of quantum physics started in the 1980's in the works of Benioff [6] and Feymann [25], who introduced the concept of *Quantum Turing Machine* (QTM), later formalized by Deutsch in 1985 [23]. This computational paradigm is based on fundamental principles of quantum mechanics, such as the *superposition* of states, the *linearity* of the evolution operators, the effects of *interference* and the principle of *observation*.

The first natural question is a comparison between the behavior of quantum models and the one of deterministic devices. From the computational power point of view, both UTM and QTM compute the same class of functions (partial recursive functions). However, the quantum model turns out to be more efficient: while the classical bit, at any given instant, can only assume one out of the possible two values, the *qbit*, which constitutes the elementary unit of information in quantum computing, can be in any linear superposition of the two fundamental states, having complex coefficients called *amplitudes*. This implies that, while a register with k bits can assume only one of the 2^k possible values, a register with k qbits can encode the information of 2^k states simultaneously, thus performing an exponential number of computations in parallel (this operation is known as *quantum parallelism*). On the other hand, while recovering a value from a classical registry does not change its content, the observation of a quantum register has the effect of making the qbits "collapse" in one of the fundamental states with a certain probability, which is related to the amplitude in the superposition. Therefore the "downside" of the quantum model is the stochastic nature of its output.

Although we can hardly expect to see a full featured quantum computer in the near future, it is reasonable to think of classical computers incorporating small quantum components. Quantum finite automata (QFAs, for short) are computational devices particularly interesting, since they represent a theoretical model for a quantum computer with finite memory [29]. QFAs exhibit both advantages and disadvantages with respect to their classical (deterministic or probabilistic) counterparts. On the one hand, quantum superposition offers some computational advantages with respect to probabilistic superposition. On the other hand, quantum dynamics are reversible: because of limitation of memory, it is sometimes impossible to simulate deterministic automata by quantum automata. Limitations due to reversibility can be partially attenuated by systematically introducing measurements of suitable observables as computational steps.

Several models of quantum automata have been proposed in the literature [29, 30]. In this thesis, we will consider the following variants (even for quantum

devices, isolated cut point acceptance will always be considered):

- *Measure-once* one-way QFAs (MO-QFAs) [7, 21, 45]: this is the simplest model of QFA, where the probability of accepting strings is evaluated by "observing" just once, at the end of input processing. The computational power of MO-QFAs is weaker than that of classical finite automata. In fact, in [7, 21, 45] it is proved that they recognize exactly the class of group languages [48], a proper subclass of regular languages.

- *Measure-many* one-way QFAs (MM-QFAs) [3, 37]: in this model, the observation is performed after each move. MM-QFAs are proved to have a recognition power stronger than MO-QFAs, but still weaker than classical finite automata [37]. Many efforts have been devoted to characterize the class of languages recognized by MM-QFAs, and this seems a really difficult task.

- QFAs *with control language* (QFCs) [10]: this is a hybrid version of QFA, "enhanced" with a classical control on the results of the step-by-step observables. This is the most powerful of the three variants: its computational power is the same as DFAs [39], and it is able to simulate any n-state MM-QFA by using n quantum states and a constant number of classical states [10].

Thesis contribution and chapter organization

In this thesis, we study some problems on classical and quantum automata working on a unary input alphabet. The central issue of this work is the *descriptional complexity* of the different models on families of languages defined through periodicity conditions on the length of the input. Descriptional complexity is a measure of succinctness for the description of an object: in the case of automata, it estimates how efficient the device is in terms of a structural resource, such as the number of states (*state complexity*) or the number of transitions (*transition complexity*). For a survey on descriptional complexity see, e.g., [32]. In this thesis we only consider state complexity, and deal with two central problems on this subject:

- given a formal language, determining how succinct a model can be, i.e., how many states are necessary for recognizing that language, and

- given two different models, establishing the conversion cost, i.e., how much

increase in the number of states is required for converting one model into the other.

In **Chapter 1** we fix the mathematical notation and give an overview of the basic principles and formalism for linear algebra, quantum mechanics and formal languages.

The following three chapters are devoted to the study of unary classical finite automata. In **Chapter 2** we give the formal definition of the three models we consider: DFAs, NFAs and PFAs. Then we discuss their behavior on unary languages. In **Chapter 3** we present a normal form for unary PFAs [18, 19], which extends the Chrobak normal form for NFAs and guarantees minimality on periodic languages. In **Chapter 4** we use this probabilistic normal form to obtain descriptional complexity results: we analyze several families of unary languages, characterized by periodicity conditions. We show that, for some of those families, all classical models require the same number of states while, for some other families, PFAs can be smaller than NFAs (sometimes reaching the theoretical lower bound), which in turn can be smaller than DFAs [14, 15].

In the last part of the thesis we focus on the quantum paradigm: in **Chapter 5** we introduce the three models of quantum automata: MO-QFAs, MM-QFAs and QFCs. Then we discuss their computational power, providing an explicit construction for MM-QFAs to recognize any unary regular language. In **Chapter 6** we focus on the descriptional complexity of QFAs: first, we present families of unary languages for which MM-QFAs require an exponentially smaller number of states with respect to their deterministic equivalent [16, 17]. Then we prove that this is very close to the (asymptotically) biggest size gap we can achieve between the two models, by showing a more general conversion lower bound on the number of states required by a DFA to simulate a QFC working on an alphabet of arbitrary size. Finally, in **Chapter 7**, we discuss periodicity problems on the behavior of MM-QFAs, presenting polynomial algorithmic solutions [16, 17].

Chapter 1

Preliminaries

In this section we set the notation and recall fundamental concepts of arithmetic and linear algebra that we will use throughout the thesis. We also introduce the principles of quantum mechanics necessary for formalizing a mathematical model of a quantum device. Finally, we recall the basic definitions of formal language theory and Chomsky's hierarchy of grammars. For more details about these three topics, see, e.g., [51, 52], [28], and [33], respectively.

The sets of natural, integer, rational, real and complex numbers are denoted respectively by $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ and $\mathbb{C}$. A *complex number* $z \in \mathbb{C}$ is defined as $\alpha = x + iy$, where $x, y \in \mathbb{R}$ and $i = \sqrt{-1}$ is the imaginary unity; we call $\mathrm{Re}(z) = x$ the *real part* and $\mathrm{Im}(z) = y$ the *imaginary part* of z. Moreover we denote by $|z| = |\sqrt{x^2 + y^2}|$ its *modulus* and by $z^* = x - iy$ its *complex conjugate*. For a set S, $|S|$ denotes its cardinality. For $x \in \mathbb{R}$ we let $\lceil x \rceil = \min\{n \in \mathbb{Z} \mid n \geq x\}$ and $\lfloor x \rfloor = \max\{n \in \mathbb{Z} \mid n \leq x\}$. Given $h, l \in \mathbb{N} \setminus \{0\}$, we let $\gcd(h, l)$ be their greatest common divisor and $\mathrm{lcm}(h, l)$ their least common multiple. We let $\langle h \rangle_l = h - l \left\lfloor \frac{h}{l} \right\rfloor$ denote the remainder of the division of h by l. For integers $h, k, l \in \mathbb{N}$, we say that $h \equiv k \pmod{l}$ if $\langle h \rangle_l = \langle k \rangle_l$. If $\langle h \rangle_l = 0$, we write $l \mid h$, otherwise we write $l \nmid h$.

We recall the following well-known results

Theorem 1.0.1 (Chinese Remainder Theorem)**.** *Given an arbitrary sequence of pairwise coprime positive integers $n_1, n_2, \ldots, n_k \in \mathbb{N}$, for any sequence of integers $a_1, a_2, \ldots, a_k \in \mathbb{Z}$, there exists an integer x solving the following system of*

simultaneous congruences:

$$\begin{aligned} x &\equiv a_1 (\text{mod } n_1) \\ x &\equiv a_2 (\text{mod } n_2) \\ &\vdots \\ x &\equiv a_k (\text{mod } n_k). \end{aligned}$$

Theorem 1.0.2 (Fundamental Theorem of Arithmetic)**.** *Any $d \in \mathbb{N} \setminus \{0\}$ admits a* factorization *as*

$$d = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n},$$

for primes $p_1, p_2, \ldots, p_n$ and $k_1, k_2, \ldots, k_n \in \mathbb{N} \setminus \{0\}$, which is unique except for the order in which the primes occur.

Theorem 1.0.3. *Let $a, b \in \mathbb{Z}$, such that they are not both zero, and let $d = \gcd(a, b)$. Then the equation*

$$ax + by = d$$

has infinite integer solutions for x, y.

1.1 Linear algebra

1.1.1 Vector spaces

A *vector space* over a field F is a set V together with two binary operations that satisfy a set of properties, listed below. Elements of V are called *vectors*, while elements of F are called *scalars*. The first operation is called *vector addition*, it takes any two vectors $v, w \in V$ and maps them into a third vector $v + w$, called the *sum of v and w*. The second operation, called *scalar multiplication*, takes any scalar $a \in F$ and any vector $v \in V$ and gives another vector av. This operation has the effect of rescaling the vector v by a factor a. To qualify as a vector space, the set V and the operations of addition and multiplication must adhere to a list of requirements called *axioms*. For any vectors $u, v, w \in V$ and scalars $a, b \in F$, the following axioms hold:

- *Associativity of addition*, i.e., $u + (v + w) = (u + v) + w$.
- *Commutativity of addition*, i.e., $u + v = v + u$.
- *Identity element of addition*, i.e., there exists an element $0 \in V$ called *zero vector*, such that $v + 0 = v$ for any $v \in V$.

- *Inverse element of addition*, i.e., there exists an element $-v \in V$, called *additive inverse of* v, such that $v + (-v) = 0$.
- *Distributivity of scalar multiplication with respect to vector addition*, i.e., $a(u + v) = au + av$.
- *Distributivity of scalar multiplication with respect to field addition*, i.e., $(a + b)v = av + bv$.
- *Compatibility of scalar multiplication with field multiplication*, i.e., $a(bv) = (ab)v$.
- *Identity element of scalar multiplication*, i.e., there exists an element $1 \in F$, such that $1v = v$.

We call a *subspace of* V any subset of V which is still a vector space.

We define the n-dimensional *real vector space* $\mathbb{R}^n$ as the set of (row) vectors $v = (a_1, a_2, \ldots, a_n)$ such that $a_1, a_2, \ldots, a_n \in \mathbb{R}$. Given a vector $v \in \mathbb{R}^n$, $(v)_j$ denotes the j-th component of v, and v^T the *transpose* (column) vector. If a vector $v \in \mathbb{R}$ has entries in the interval $[0, 1]$ and it holds $\sum_{j=1}^{n}(v)_j = 1$, we say that v is a *stochastic vector*. We denote by e_k the boolean row vector such that $(e_k)_j = 1 \Leftrightarrow j = k$.

The n-dimensional *complex vector space* $\mathbb{C}^n$ is the set of (row) vectors $\varphi = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, with $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{C}$. The *transposed conjugate* of φ is $\varphi^\dagger = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*)^T$.

Given $\varphi_1, \varphi_2, \ldots, \varphi_m \in \mathbb{C}^n$ and $a_1, a_2, \ldots, a_m \in \mathbb{C}$, we say that the vector

$$\varphi = a_1\varphi_1 + a_2\varphi_2 + \ldots + a_m\varphi_m$$

is a *linear combination* of the m vectors. We say that $\varphi_1, \varphi_2, \ldots, \varphi_m$ are *linearly independent* if it holds

$$\begin{gathered} a_1\varphi_1 + a_2\varphi_2 + \ldots + a_m\varphi_m = 0 \\ \Updownarrow \\ a_1 = a_2 = \ldots = a_m = 0. \end{gathered}$$

A set of n independent vectors in $\mathbb{C}^n$ is called a *base* of the vector space $\mathbb{C}^n$; every vector in $\mathbb{C}^n$ can be univocally obtained as a linear combination of the base. Clearly, the above definitions also hold for the real case.

A vector space (over the field of complex numbers) is an *inner product space* when it is equipped with an operation $\langle , \rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ called *inner product* such that, for any vectors $x, y, z \in \mathbb{C}^n$ and scalar $\lambda \in \mathbb{C}$, the following properties hold

- $\langle x, y\rangle = \langle y, x\rangle^*$,
- $\langle x + y, z\rangle = \langle x, z\rangle + \langle y, z\rangle$ and $\langle x, y + z\rangle = \langle x, y\rangle + \langle x, z\rangle$,
- $\langle \lambda x, y\rangle = \lambda\langle x, y\rangle$,
- $\langle x, x\rangle$ is a nonnegative real number, and $\langle x, x\rangle = 0$ only if x is the zero vector.

In the vector space $\mathbb{R}^n$, the inner product of two vectors $v = (a_1, a_2, \ldots, a_n)$ and $u = (b_1, b_2, \ldots, b_n)$ is

$$\langle v, u\rangle = vu^T = \sum_{j=1}^{n} a_j b_j.$$

The introduction of the inner product allows us to formally define the concept of length of a vector: the inner product of a vector with itself has a particular geometrical value, in fact, by Pythagoras' Theorem, $\langle v, v\rangle = \sum_{j=1}^{n} (v)_j^2$ is exactly the square of the length of the vector v. Therefore, we say that the *length* (or *ℓ_2-norm*) of a vector v is $\|v\| = \sqrt{\langle v, v\rangle}$, and it satisfies the following properties

- $\|x\| \geq 0$, and equality holds only if $x = 0$,
- $\|\lambda x\| = |\lambda|\|x\|$,
- $\|x + y\|^2 = \|x\|^2 + \|y\|^2 + \langle x, y\rangle + \langle y, x\rangle$,
- (*parallelogram law*) $\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2$,
- (triangle inequality) $\|x + y\| \leq \|x\| + \|y\|$,
- (Cauchy-Schwarz inequality) $|\langle x, y\rangle| \leq \|x\|\|y\|$.

The ℓ_2-norm determines a distance function over the elements of V, defined as

$$d(x, y) = \|x - y\|$$

and called *Euclidean metric*. The space $\mathbb{R}^n$ with the Euclidean metric is called *Euclidean space*. We denote the *ℓ_1-norm* of v by $\|v\|_1 = \sum_{j=1}^{n} |(v)_j|$.

The inner product also allows us to define the angle between two vectors $v, u \in \mathbb{R}^n$ as

$$\operatorname{ang}(v, u) = \arccos\left(\frac{\langle v, u\rangle}{\|v\|\|u\|}\right). \tag{1.1}$$

For complex vectors, we need a slightly different definition of inner product, in order to avoid that vectors have complex or negative lengths. Therefore, we

define the inner product of two complex vectors $\varphi = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $\psi = (\beta_1, \beta_2, \ldots, \beta_n)$ as

$$\langle \varphi, \psi \rangle = \varphi \psi^\dagger = \sum_{j=1}^{n} \alpha_j \beta_j^*,$$

so that the length function satisfies the same properties defined for the real case. In particular, $\|\varphi\| = \sqrt{\langle \varphi, \varphi \rangle}$ is a real value greater than 0, because for every complex number α, it holds that $\alpha \cdot \alpha^*$ is a real positive value. We say that $\varphi \in \mathbb{C}$ is a *unitary vector* if $\|\varphi\| = 1$. Since the inner product between two complex vectors is, in general, a complex number, the angle between $\varphi, \psi \in \mathbb{C}$ is defined as

$$\text{ang}(\varphi, \psi) = \arccos\left(\frac{\text{Re}(\langle \varphi, \psi \rangle)}{\|\varphi\|\|\psi\|}\right).$$

Two vectors $v, w \in V$ in an inner product space V are said to be *orthogonal* to each other ($v \perp w$) if it holds $\langle \varphi, \psi \rangle = 0$. Moreover, if $S, T \subseteq V$, we say that S *is orthogonal to* T ($S \perp T$) if, for all $v \in S$ and $w \in T$, it holds $v \perp w$. A set $S \subseteq V$ is called *orthonormal* if all its members are unitary and pairwise orthogonal.

We say that a sequence $\{v_k\}$ of vectors in V *converges in norm* to a vector $y \in V$ if and only if $\lim_{n\to\infty} \|x_n - y\| = 0$, and it is a *Cauchy sequence* if and only if, for every $\varepsilon > 0$ there is an integer $N_\varepsilon \in \mathbb{N}$ such that, for all $n, m > N_\varepsilon$, $\|x_n - x_m\| < \varepsilon$. An inner product space V is *complete* if and only if every Cauchy sequence in V converges in norm to some vector in V. A complete inner product space is called *Hilbert space*.

For a vector $p \in \mathbb{R}^n$ and a real positive value $r > 0$, we call $\mathcal{B}_r(p) = \{v \in \mathbb{R}^n \mid \|v - p\| \leq r\}$ the *ball of radius r centered in p*, and we call $\mathcal{B}_r^<(p) = \{v \in \mathbb{R}^n \mid \|v - p\| < r\}$ the *open ball of radius r centered in p*. We say that $S \subseteq \mathbb{R}^n$ is *totally bounded* if and only if for every real number $\varepsilon > 0$, there exists a finite collection of open balls in S of radius ε whose union contains S. Any subset of the Euclidean space which is complete and totally bounded is called *compact*. An example of compact space is the unitary ball $\mathcal{B}_1(0)$ in $\mathbb{R}^n$.

1.1.2 Matrices

Real matrices

A square matrix $M \in \mathbb{R}^{n\times n}$ is said to be of *order n*, and can be seen as a linear transformation on the vector space $\mathbb{R}^n$:

$$M : \mathbb{R}^n \to \mathbb{R}^n.$$

We denote by I_{nxn} the $n \times n$ identity matrix[1]. For a square matrix M we call $\det(M)$ its determinant. For matrices $M \in \mathbb{R}^{m \times n}$ and $M' \in \mathbb{R}^{p \times q}$ and $a \in \mathbb{R}$, we will denote by

- $(M)_{j,k}$ the element in (j,k)-th position of M,
- $M + M'$ the sum of M and M', defined when $m = p$ and $n = q$ as $(M + M')_{j,k} = (M)_{j,k} + (M)_{j,k}$,
- MM' the matrix product of M and M', defined when $n = p$, as $(MM')_{j,k} = \sum_{t=1}^{n} (M)_{j,t}(M')_{t,k}$,
- aM the scalar multiplication of a and M, defined as $(aM)_{j,k} = a(M)_{j,k}$,
- $M^{-1} \in \mathbb{R}^{n \times m}$ the inverse matrix of M, whenever it exists, which is such that $MM^{-1} = M^{-1}M = I$,
- $M^T \in \mathbb{R}^{n \times m}$ the transpose matrix, defined as $(M^T)_{j,k} = (M)_{k,j}$,
- $\|M\|_1 = \max\{\|vM\|_1 \mid v \in \mathbb{R}^n \text{ and } \|v\|_1 \leq 1\}$ the *ℓ_1-norm* of M.

The *characteristic equation* of a square matrix M is the equation in one variable λ

$$\det(M - \lambda I) = 0,$$

where det denotes the determinant operation.

A real matrix M is said to be *(sub)stochastic* whenever its entries are from the interval $[0, 1]$, and each row sum is (less than or) equal to 1. One may easily verify that

Lemma 1.1.1. *For any (sub)stochastic matrix $M \in \mathbb{R}^{n \times m}$ and any matrix $A \in \mathbb{R}^{p \times q}$, the following properties hold:*

1. *if $m = p$, then $\|MA\|_1 \leq \|A\|_1$,*
2. *if $q = n$, then $\|AM\|_1 \leq \|A\|_1$.*

Proof. Since M is (sub)stochastic, it holds $\|vM\|_1 \leq \|v\|_1$, for an vector $v \in \mathbb{R}^n$. By calling $x = \arg\max_{\{v \in \mathbb{R}^n \mid \|v\|_1 \leq 1\}} \|vMA\|_1$, we have that

$$\|MA\|_1 = \|xMA\|_1 \leq \|xA\|_1,$$

and since $\|x\| \leq 1$, it holds $\|xA\|_1 \leq \|A\|_1$, so Property (1) is proved.

[1] when the order n is clear from the context, we will sometimes write just I

On the other hand, by calling $y = \arg\max_{\{v\in\mathbb{R}^n \mid \|v\|_1\leq 1\}} \|vAM\|_1$, it holds that

$$\|AM\|_1 = \|yAM\| \leq \|yA\|,$$

then again, since $\|y\| \leq 1$, it holds $\|yA\|_1 \leq \|A\|_1$, so Property (2) is also proved. ∎

A set S of indices for a stochastic matrix M is called an *ergodic class* if and only if:

- for every $i, j \in S$ there exists $h \in \mathbb{N}$ such that $(M^h)_{i,j} > 0$, and
- for every i, j such that $i \in S$ and $j \notin S$ it holds $(M)_{i,j} = 0$.

The *period* of the ergodic class S in a matrix M of order n is defined as

$$k = \gcd\{h \leq n \mid (M^h)_{j,j} > 0, j \in S\}.$$

A matrix M is said to be in *canonical form* if and only if the indices of the same ergodic class form a set of consecutive integers, and they are smaller than the indices which do not belong to any ergodic class. More formally, a matrix M in canonical form with m ergodic classes has the following structure:

$$M = \begin{pmatrix} M_1 & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & M_m & 0 \\ T_1 & T_2 & \dots & T_m & T_{m+1} \end{pmatrix} \tag{1.2}$$

where, for $1 \leq j \leq m$, M_j is the square matrix representing the j-th ergodic class while, for $1 \leq j \leq m+1$, T_j is some matrix, not necessarily square. If M is stochastic, then all M_j's are stochastic matrices and all T_j's are substochastic matrices. It is well-know that, by performing a suitable index permutation, any stochastic matrix can be put in canonical form. Thus, without loss of generality, we may assume our stochastic matrices to be in canonical form.

Theorem 1.1.2. *For any stochastic matrix M in canonical form with m ergodic classes, as in (1.2), there exist $l_1, \dots, l_m \in \mathbb{N} \setminus \{0\}$ such that every l_j does not exceed the order of M_j, and $\lim_{q\to\infty} \left(M_j{}^{l_j}\right)^q$ exists. Moreover, $\lim_{q\to\infty}(T_{m+1})^q = 0$.*

Actually, every l_j in Theorem 1.1.2 coincides with the period of the ergodic class represented by M_j. The following Lemma describes the powers of a square stochastic matrix in canonical form:

Lemma 1.1.3. *Let M be a stochastic matrix in canonical form as in (1.2), with l_j being the period of the ergodic class represented by M_j, for every $1 \leq j \leq m$. Given $h \in \mathbb{N}$, we have*

$$M^h = \begin{pmatrix} M_1^h & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & M_m^h & 0 \\ \hat{T}_{1,h} & \hat{T}_{2,h} & \cdots & \hat{T}_{m,h} & T_{m+1}^h \end{pmatrix}, \tag{1.3}$$

for suitable matrices $\hat{T}_{1,h}, \ldots, \hat{T}_{m,h}$. Moreover, for every $1 \leq j \leq m$ and every $0 \leq r_j < l_j$, $\lim_{q\to\infty} \hat{T}_{j,ql_j+r_j}$ exists.

Proof. It is sufficient to prove that, for any $\varepsilon > 0$, there exists $H \in \mathbb{N}$ such that for every $q > 0$ and every $0 \leq r_j < l_j$ we have $\|\hat{T}_{j,Hl_j+r_j} - \hat{T}_{j,(H+q)l_j+r_j}\|_1 \leq \varepsilon$, implying that $\{\hat{T}_{j,ql_j+r_j}\}_{q\in\mathbb{N}}$ is a Cauchy sequence. Set $H = N_1 + N_2$ such that

$$\left\|(\hat{T}_{m+1})^{N_1 l_j + r_j} - 0\right\|_1 \leq \frac{\varepsilon}{4} \quad \text{and} \quad \left\|M_j^{N_2 l_j} - M_j^{(N_2+q)l_j}\right\|_1 \leq \frac{\varepsilon}{2}. \tag{1.4}$$

Notice that such N_1 and N_2 always exist by Theorem 1.1.2. Now, compute $\hat{T}_{j,Hl_j+r_j}$ by performing the product $M^{N_1 l_j + r_j} \cdot M^{N_2 l_j}$ to get

$$\hat{T}_{j,Hl_j+r_j} = \hat{T}_{j,N_1 l_j+r_j} \cdot M_j^{N_2 l_j} + (T_{m+1})^{N_1 l_j + r_j} \cdot \hat{T}_{j,N_2 l_j}.$$

Similarly, by the product $M^{N_1 l_j + r_j} \cdot M^{(N_2+q)l_j}$, compute $\hat{T}_{j,(H+q)l_j+r_j}$ as

$$\hat{T}_{j,(H+q)l_j+r_j} = \hat{T}_{j,N_1 l_j+r_j} \cdot M_j^{(N_2+q)l_j} + (T_{m+1})^{N_1 l_j + r_j} \cdot \hat{T}_{j,(N_2+q)l_j}.$$

Since all matrices $\hat{T}_{j,h}$ are substochastic, by Lemma 1.1.1 and (1.4), we get

$$\begin{aligned} \left\|\hat{T}_{j,Hl_j+r_j} - \hat{T}_{j,(H+q)l_j+r_j}\right\|_1 = &\left\|\hat{T}_{j,N_1 l_j+r_j} \cdot M_j^{N_2 l_j} + (T_{m+1})^{N_1 l_j + r_j} \cdot \hat{T}_{j,N_2 l_j} + \right. \\ &\left. -\hat{T}_{j,N_1 l_j+r_j} \cdot M_j^{(N_2+q)l_j} - (T_{m+1})^{N_1 l_j + r_j} \cdot \hat{T}_{j,(N_2+q)l_j}\right\|_1 \\ \leq &\left\|\hat{T}_{j,N_1 l_j+r_j} \cdot \left(M_j^{N_2 l_j} - M_j^{(N_2+q)l_j}\right)\right\|_1 + \\ &+ \left\|(T_{m+1})^{N_1 l_j + r_j} \cdot \hat{T}_{j,N_2 l_j}\right\|_1 + \\ &+ \left\|(T_{m+1})^{N_1 l_j + r_j} \cdot \hat{T}_{j,(N_2+q)l_j}\right\|_1 \\ \leq &\frac{\varepsilon}{2} + \frac{\varepsilon}{4} + \frac{\varepsilon}{4}. \end{aligned}$$

∎

By following notations in Lemma 1.1.3, let us set $\lim_{q\to\infty} \hat{T}_{j,ql_j+r_j} = \hat{T}_{j,r_j}^{\infty}$, for every $0 \leq r_j \leq l_j - 1$. With a slight abuse of terminology, Lemma 1.1.3 enables

us to say that the sequence $\{\hat{T}_{j,h}\}_{h\in\mathbb{N}}$ has the *periodic limit* $\hat{T}^{\infty}_{j,0},\dots,\hat{T}^{\infty}_{j,l_j-1}$ in the following sense: For any $\varepsilon>0$, there exists $H\in\mathbb{N}$ such that for any $h>H$ we have $\|\hat{T}_{j,h}-\hat{T}^{\infty}_{j,\langle h\rangle_{l_j}}\|_1<\varepsilon$.

A stochastic matrix consisting of only a single ergodic class is called *irreducible*. Clearly, any irreducible matrix is in canonical form. On the other hand, given a stochastic matrix M in canonical form, as in (1.2), each submatrix M_j is irreducible. Even for an irreducible stochastic matrix, we have the existence of a periodic limit for its powers:

Theorem 1.1.4. *Let P be an irreducible stochastic matrix of period $\ell\in\mathbb{N}\setminus\{0\}$. Then $\lim_{q\to\infty}P^{q\ell+r}$ exists for any $0\le r<\ell$.*

Complex matrices

For a complex matrix $V\in\mathbb{C}^{m\times n}$ we use the same notation as the one for real matrices, moreover:

- V^* is the conjugate matrix, defined as $(V^*)_{i,j}=(V)^*_{i,j}$,
- $V^\dagger$ is the adjoint matrix, defined as $V^\dagger=(V^T)^*$.

For a complex matrix V of order n and a complex number λ, the following statements are equivalent:

1. λ is an eigenvalue of V.
2. λ is the solution of the characteristic equation of V, i.e., $\det(V-\lambda I)=0$.
3. There is a nonzero vector $\varphi\in\mathbb{C}^n$ such that $\varphi V=\lambda\varphi$.

If λ is an eigenvalue of a matrix $V\in\mathbb{C}^{n\times n}$, the *eigenspace* of V associated to λ is the set

$$S(V,\lambda)=\{\varphi\in\mathbb{C}^n\mid\varphi V=\lambda\varphi\}.$$

The nonzero vectors in $S(V,\lambda)$ are the *eigenvectors* of V associated with λ. We can extend the concept of norm to matrices, by defining $\|V\|=\max_{\|\varphi\|=1}\|\varphi V\|$. Two matrices $V_1,V_2\in\mathbb{C}^{n\times n}$ are *similar* whenever there exists an invertible matrix $X\in\mathbb{C}^{n\times n}$ satisfying $V_1=XV_2X^{-1}$. Similar matrices have the same characteristic equation.

A complex square matrix V is *unitary* if it holds

$$VV^\dagger=V^\dagger V=I.$$

For unitary matrices, we have the following properties:

- The eigenvalues and the determinant of a unitary matrix are complex numbers with unitary modulus,
- unitary matrices preserve the norm, i.e. V is unitary iff for any $\varphi \in \mathbb{C}^n$, $\|\varphi\| = \|\varphi V\|$,
- if V_1 and V_2 are unitary matrices, then so are $V_1^{-1}, V_1^*, V_1V_2, V_1 \oplus V_2$ and $V_1 \otimes V_2$, where $\oplus$ and $\otimes$ denote the direct sum and product introduced in Section 1.1.3.

A complex matrix V is *Hermitian* if it holds

$$V = V^\dagger.$$

For every Hermitian matrix H there exists a unitary matrix U such that

$$H = U\Lambda U^{-1},$$

where Λ is a diagonal matrix. Moreover, vor every $n \geq 0$, it holds

$$H^n = U\Lambda^n U^{-1}.$$

Projectors

In the two-dimensional space $\mathbb{R}^2$ the *projection operator* (or simply, *projector*) on the first Cartesian axis is the matrix

$$P_x = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

such that, for a vector $v = (a, b)$, $vP_x = (a, 0)$. The effect of P_x is to project the vector v on the subspace of $\mathbb{R}^2$ identified by the first Cartesian axis. A geometrical interpretation of this operation is shown in Figure 1.1

This can be generalized to any n-dimensional complex space: given a subspace $S \subseteq \mathbb{C}^n$, by calling $S^\perp = \{\varphi \in \mathbb{C}^n \mid \forall \psi \in S, \langle \varphi, \psi \rangle = 0\}$ the subspace of $\mathbb{C}^n$ orthogonal to S, we can decompose any vector $\varphi \in \mathbb{C}^n$ as $\varphi = \phi_1 + \phi_2$ such that $\phi_1 \in S$ and $\phi_2 \in S^\perp$. We define the projection operator P_S on the subspace S such that

$$\varphi P_S = \phi_1.$$

A matrix P is a projector if and only if it is *idempotent*, i.e. if $P^2 = P$, and Hermitian.

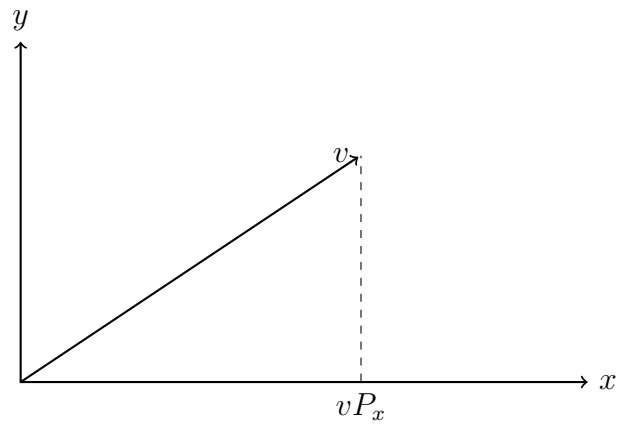


Figure 1.1: Geometrical interpretation of the projection operator.

1.1.3 Operations on Matrices

We now introduce two operations which allow us to act on several vectors with different matrices at the same time and independently. These operations are frequently used to induce combination of events on finite state automata.

Given two complex vectors $\varphi = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $\psi = (\beta_1, \beta_2, \ldots, \beta_m)$, we define the *direct sum of vectors* φ, ψ as the $n+m$-dimensional vector

$$\varphi \oplus \psi = (\varphi, \psi) = (\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_m).$$

Given two matrices $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{p \times q}$, where

$$A = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \ldots & \alpha_{mn} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} \beta_{11} & \beta_{12} & \ldots & \beta_{1q} \\ \beta_{21} & \beta_{22} & \ldots & \beta_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \ldots & \beta_{pq} \end{pmatrix},$$

we define the *direct sum of matrices* A and B as the $(m+p) \times (n+q)$ matrix

$$A \oplus B = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \ldots & \alpha_{1n} & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \ldots & \alpha_{mn} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \beta_{11} & \ldots & \beta_{1q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & \beta_{p1} & \ldots & \beta_{pq} \end{pmatrix}.$$

Properties:

The direct sum is often used to parallelize the application of different matrices on different vectors, since it holds

$$(\varphi \oplus \psi)(A \oplus B) = \varphi A \oplus \psi B.$$

In other words, the direct sum of vectors φA and ψB, obtained with two different matrix applications, can be obtained in a single step by applying the operator $A \oplus B$ to the vector $\varphi \oplus \psi$. Moreover, for matrices A, B, C, D it holds:

$$(A \oplus B)(C \oplus D) = AC \oplus BD. \tag{1.5}$$

For two vectors $\varphi = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $\psi = (\beta_1, \beta_2, \ldots, \beta_m)$, we define the *Kronecker product of vectors* φ, ψ as the nm-dimensional vector

$$\varphi \otimes \psi = (\alpha_1\psi, \alpha_2\psi, \ldots \alpha_n\psi) = (\alpha_1\beta_1, \alpha_1\beta_2, \ldots, \alpha_1\beta_n, \ldots, \alpha_n\beta_1, \alpha_n\beta_2, \ldots, \alpha_n\beta_m).$$

Given two matrices $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$, where

$$A = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \ldots & \alpha_{mn} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} \beta_{11} & \beta_{12} & \ldots & \beta_{1q} \\ \beta_{21} & \beta_{22} & \ldots & \beta_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \ldots & \beta_{pq} \end{pmatrix},$$

we define the *Kronecker product of matrices* A and B as the $(mp) \times (nq)$ matrix

$$A \otimes B = \begin{bmatrix} \alpha_{11}B & \alpha_{12}B & \ldots & \alpha_{1n}B \\ \alpha_{21}B & \alpha_{22}B & \ldots & \alpha_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1}B & \alpha_{m2}B & \ldots & \alpha_{mn}B \end{bmatrix}.$$

Properties:

Symmetrically to the direct sum, the Kronecker product of vectors φA and ψB, obtained with two different matrix applications, can be obtained in a single step by applying the operator $A \otimes B$ to the vector $\varphi \otimes \psi$. More formally:

$$(\varphi \otimes \psi)(A \otimes B) = \varphi A \otimes \psi B.$$

Moreover, for vectors φ, ψ, ϕ, ξ and matrices A, B, C, D, the following properties hold

- $\langle \varphi \otimes \psi, \phi \otimes \xi \rangle = \langle \varphi, \phi \rangle \cdot \langle \psi, \xi \rangle$,
- $(A \otimes B)(C \otimes D) = AC \otimes BD$,
- $\|\varphi \otimes \psi\| = \|\varphi\|\|\psi\|$.

1.2 Mathematical formulation of quantum mechanics

We now introduce the fundamental postulates of quantum mechanics necessary for formalizing mathematical models of quantum devices. We will define the concept of state, evolution and measurement of a quantum system, and we will give a mathematical representation of the *qbit*, which is the quantum counterpart of the classical bit.

Describability of a quantum system:

> *Every physical system can be associated with a complex Hilbert space, called the system space of states. At any given moment, the system is completely described by a state vector, which is a unitary vector in the space of states.*

This allows us to give a formal definition of a *qbit*, which represents the simplest quantum system. The qbit is the quantum analogoue of the classical bit, and it is characterized by two elementary states[2]:

$$\mathbf{0} = (1,0) \qquad \text{and} \qquad \mathbf{1} = (0,1)\,,$$

forming an orthonormal base for the Hilbert space $\mathbf{C}^2$. A generic state of the qbit is a unitary vector

$$\psi = \alpha_0\,(1,0) + \alpha_1\,(0,1)\,,$$

for some complex values α_0, α_1 such that the unitary constraint $|\alpha_0|^2 + |\alpha_1|^2 = 1$ holds.

Evolution of a quantum system in discrete time:

> *The evolution of a (discrete time) quantum system is described by a unitary transformation: by calling ψ the state of the system at time t and ξ the state of the system at time $t+1$, it must hold*
>
> $$\xi = \psi U,$$
>
> *for some unitary matrix U.*

[2]Qbits are usually represented with the Dirac notation. Here we present an equivalent definition of qbit in terms of row vectors, in order to have a notation consistent with the rest of the thesis.

In other words, the evolved configuration of the system is obtained by applying U to the current state. The fact that U is a unitary matrix guarantees the reversibility of the operation: from ξ one can reconstruct the state ψ through the adjoint matrix $\mathrm{U}^\dagger$:

$$\psi = \xi U^\dagger.$$

In continuous time, the evolution of a quantum system can be described by Schrödinger's equation in terms of Hamiltonians. However, we will not present it in this thesis, since we will only deal with quantum systems evolving in discrete time.

Measurement of a quantum system:

> *An observable $\mathcal{O}$ of a quantum system, i.e. any measurable property of a physical system, is represented by a Hermitian operator on the observed space. $\mathcal{O}$ is usually represented by its so-called* spectral decomposition
>
> $$\mathcal{O} = \sum_{i=1}^{q} \nu_i P_i,$$
>
> *where P_i is the projection operator on the subspace of the eigenvectors of $\mathcal{O}$ corresponding to the eigenvalue ν_i.*

The possible outcomes of a measurement on the observable $\mathcal{O}$ are all the eigenvalues of $\mathcal{O}$: after measuring a quantum system in the state ψ, the probability of obtaining the eigenvalue ν_i as an outcome is

$$p(\nu_i) = \|\psi P_i\|^2.$$

The operators P_i satisfy the completeness equation

$$\sum_{i=1}^{q} P_i = I$$

which guarantees that the total probability of the measurement outcomes is 1. In fact, since all P_i's are orthogonal, we can write

$$1 = \sum_{i=1}^{q} p(\nu_i) = \sum_{i=1}^{q} \|\psi P_i\|^2 = \|\psi \sum_{i=1}^{q} P_i\|^2 = \|\psi\|^2.$$

Unlike the classical computation, the process of quantum observation does not leave the system unchanged: after a measurement on ψ with outcome ν_i, the system new state is

$$\frac{\psi P_i}{\sqrt{p(\nu_i)}}.$$

This type of measurement is called *projective measurement*, since the observable $\mathcal{O}$ is described by an arbitrary set of orthogonal projectors satisfying the completeness equation.

1.3 Formal languages

We define an *alphabet* as a (finite) set of symbols $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. A *word* ω on the alphabet Σ is any finite sequence of symbols belonging to Σ, and we denote by $|\omega|$ the *length* of ω. The word of length zero is called the *empty word* and will be denoted by ϵ. We call Σ^* the set of all words on Σ and we call $\Sigma^{<n}$ ($\Sigma^{>n}$) the set of words on Σ of length smaller (greater) than n.

A *formal language* is any subset of Σ^*. For languages $L_1, L_2 \subseteq \Sigma^*$, we recall that

- the *union* of L_1 and L_2 is the language $L_1 \cup L_2 = \{\omega \in \Sigma^* \mid \omega \in L_1 \vee \omega \in L_2\}$,
- the *intersection* of L_1 and L_2 is the language $L_1 \cap L_2 = \{\omega \in \Sigma^* \mid \omega \in L_1 \wedge \omega \in L_2\}$,
- the *product* of L_1 and L_2 is the language $L_1 \cdot L_2 = \{\omega = xy \in \Sigma^* \mid x \in L_1 \wedge y \in L_2\}$,
- the *j-th power* of L_1 is the language $L_1^j = L_1 \cdot L_1^{j-1}$, and $L^0 = \{\epsilon\}$,
- the *complement* of L_1 is the language $L_1^c = \{\omega \in \Sigma^* \mid \omega \notin L_1\}$,
- the *Kleene closure* of L_1 is the language $L_1^* = \bigcup_{j=0}^{\infty} L_1^j$.

We call $\chi_L : \Sigma^* \to \{0, 1\}$ the *characteristic function* of a language L, if it holds

$$\chi_L(\omega) = \begin{cases} 1 & \text{if } \omega \in L \\ 0 & \text{if } \omega \notin L. \end{cases}$$

For reprensenting a (potentially infinite) formal language L with a finite amount of information, there are two main approaches: L can be seen as the result of a generative system (e.g., grammar) or as the set of words recognized by an accept/reject device (automaton).

From a mathematical point of view, a *grammar* is a structure represented by a tuple

$$G = \langle \Sigma, Q, P, S \rangle,$$

where Σ is the alphabet of symbols composing the words in the generated language, Q is the set of *variables*, P is a finite set of *production rules*, and $S \in Q$

is a special variable called *axiom*. The production rules are of the form $\alpha \to \beta$, where $\alpha \in (\Sigma \cup Q)^+$ and $\beta \in (\Sigma \cup Q)^*$. Moreover, for $a, b \in (\Sigma \cup Q)^*$, we say that *a generates b in one step* ($a \Rightarrow b$) if $a = fxg, b = fyg$, for $f, g, x, y \in (\Sigma \cup Q)^*$, and $x \to y$ is a rule in P. In general, we say that *a generates b* ($a \Rightarrow^* b$) if either

(i) $a = b$ or

(ii) there exists $c \in (\Sigma \cup Q)^*$ such that $a \Rightarrow c$ and $c \Rightarrow^* b$.

Different types of rules define different types of grammars, classified by Chomsky as follows:

- *Type* 0 *grammars:* arbitrary production rules.
- *Type* 1 *grammars:* production rules are of the form $\alpha \to \beta$, with $|\alpha| \leq |\beta|$; the rule $S \to \epsilon$ is permitted as long as S does not appear on the right side of any rule.
- *Type* 2 *grammars:* production rules are of the form $A \to \beta$, with $A \in Q$.
- *Type* 3 *grammars:* production rules are of the form $A \to \sigma B$, $A \to \sigma$ and $A \to \epsilon$, with $A, B \in Q$ and $\sigma \in \Sigma$.

Grammars are used to generate formal languages: for a grammar $\langle \Sigma, Q, P, S \rangle$, the *language generated* by G is

$$L_G = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}.$$

Languages also have a classification, which depends on the type of the generating grammar. More precisely, a language is *of type* k (with $k \in \{0, 1, 2, 3\}$) if it can be generated by a grammar of type k.

An *accepting system* (or *automaton*) for the language $L \subseteq \Sigma^*$ is a device which, given a word $\omega \in \Sigma^*$ as input, decides whether $\omega \in L$ or not.

Languages of type 3 (also called *regular*) are recognized by finite state automata, type 2 languages (*context free*) are recognized by nondeterministic pushdown automata, type 1 languages have linear bounded nondeterministic Turing machine as accepting system, while type 0 languages coincide with the class of *recursively enumerable* languages, accepted by Turing machines.

For the four classes of languages it holds

$$\text{Type } k \text{ languages} \subset \text{Type } k-1 \text{ languages},$$

for $k \in \{1, 2, 3\}$.

Chapter 2

Classical automata and regular languages

Finite state automata are an abstract model for many types of algorithms and computational devices. Since they provide a simplified mathematical representation of the computational dynamics, they are a powerful tool for studying characteristics and theoretical properties of a system, while ignoring the implementation details or the specific machine's hardware.

In this Section we present the formal definition of some of the most studied variants of classical finite state automata. Moreover, we introduce two important subclasses of regular languages, namely cyclic and unary.

2.1 Deterministic finite state automata

A *one way*[1] *finite state deterministic automaton* (DFA) is conceived as an abstract machine, which has an infinite *input tape* made of cells, where each cell may contain one symbol from the input alphabet. This tape is accessible through a reading head, which can only move from left to right over the input tape. The computational part of the device, called the *finite control*, is a black box which, at any given moment, is in one among a finite set of possible configurations, called *states*, which can be accepting or nonaccepting. Initially the finite control is set to a specific starting state, and the reading head is placed on the leftmost cell of the reading tape. At discrete time steps, the automaton reads one symbol from the input tape and enters a new state that depends only on the current state and the symbol just read. After reading an input symbol, the reading head moves

[1] In the literature, this model is often referred to as *real-rime*, meaning it consumes one input symbol at every step [29], while *one-way* denotes devices that never move the input head to the left, and hence can have stationary moves.

one position to the right on the input tape, so that on the next step it will read the symbol in the next tape cell. More formally, a DFA is a tuple

$$A = \langle\ \Sigma,\ Q,\ q_0,\ \tau,\ F\ \rangle$$

where:

- Σ is the (finite) *alphabet* of input symbols,
- Q is the (finite) set of *states* of the automaton,
- $q_0 \in Q$ is the *starting state*,
- τ is the *transition function* $\tau : Q \times \Sigma \to Q$,
- $F \subseteq Q$ is the set of the *accepting states*.

The function τ can be univocally extended to the set $Q \times \Sigma^*$ by defining the function τ^* as follows:

- $\tau^*(q, \epsilon) = q$ for every $q \in Q$,
- $\tau^*(q, \omega\sigma) = \tau(\tau^*(q, \omega), \sigma)$ for $q \in Q$, $\omega \in \Sigma^*$ and $\sigma \in \Sigma$.

A *computation* of A on the input word $\omega = \sigma_1\sigma_2\cdots\sigma_n$ is a sequence of states $q_0, q_1, \ldots, q_f$ such that, for every $1 \leq i \leq f$, it holds $\tau(q_{i-1}, \sigma_i) = q_i$. If q_f belongs to the set F, we say that A *accepts* the word ω. The *language recognized* by the DFA A is

$$L_A = \{\omega \in \Sigma^* \mid \tau^*(q_0, \omega) \in F\}.$$

Finite state automata are often visualized as oriented graphs (see Figure 2.1), where the vertices represent states, while edges describe the transition function, and are labelled with the symbol that triggers the transition. Accepting states are drawn with a double circle and the initial state is marked by an incoming edge.

An alternative definition of DFA with k states can be given in term of its *matrix representation*:

$$A = \langle \varphi, \{M(\sigma)\}_{\sigma\in\Sigma}, \eta \rangle,$$

where:

- $\varphi \in \{0,1\}^k$ is the *characteristic vector of the initial state*,
- $\{M(\sigma)\}_{\sigma\in\Sigma}$ is the set of *transition matrices*,
- $\eta \in \{0,1\}^k$ is the *characteristic vector of the final states*.

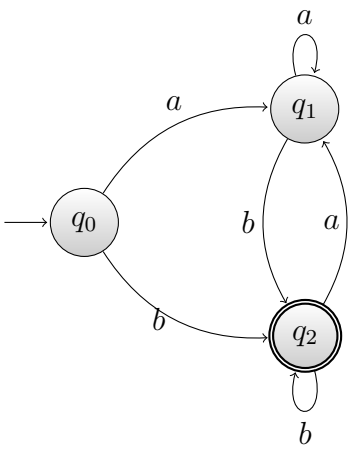


Figure 2.1: DFA over the alphabet $\{a, b\}$, recognizing the language of all words ending with the symbol b.

The representation of the states, which were previously identified by the set $Q = \{q_1, q_2, \ldots, q_k\}$, is now performed through *characteristic vectors*: the state q_i is represented by a boolean row vector having a 1 in its i-th position and 0 anywhere else. For each $0 \leq i \leq n$, the vector $\eta \in \{0, 1\}^k$ has a 1 in its i-th position iff q_i is an accepting state. The state transition previously determined by τ is now described by the square matrix $M(\sigma)$ of order k, such that $(M(\sigma))_{i,j} = 1$ if the symbol σ triggers a transition from state q_i to state q_j, otherwise $(M(\sigma))_{i,j} = 0$. Since the automaton is deterministic, for each $\sigma \in \Sigma$, the matrix $M(\sigma)$ is boolean and stochastic, i.e. each row contains a 1 in exactly one position.

As we generalized τ over the set $Q \times \Sigma^*$, for any word $\omega = \sigma_1\sigma_2 \cdots \sigma_n$, we can write $M(\omega) = M(\sigma_1)M(\sigma_2) \cdots M(\sigma_n)$. The language recognized by the DFA A in matrix form is

$$L_A = \{\omega \mid \varphi M(\omega)\eta = 1\}.$$

2.2 Nondeterministic finite state automata

Another variant of finite state automaton is obtained by considering DFAs and dropping the constraint of determinism: a *one way nondeterministic finite state automaton* (NFA) is a device similar to a DFA, which is allowed to be in more than one state at any given time.

In the graph representation (see Figure 2.2), this means that, for a given state q and symbol σ, there can be more than one outgoing edge from q labelled with σ. We also allow multiple initial states.

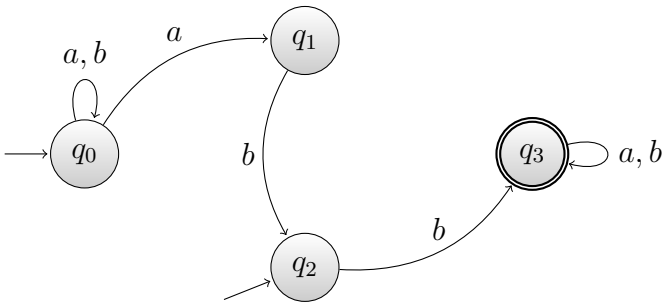


Figure 2.2: NFA over the alphabet $\{a, b\}$, accepting the language of all words containing the pattern abb or starting with the symbol b.

Like DFAs, also NFAs have a matrix representation

$$A = \langle \varphi, \{M(\sigma)\}_{\sigma \in \Sigma}, \eta \rangle,$$

in which the initial vector φ can now represent more than one state, so it is a generic boolean vector, and $M(\sigma)$ is any boolean matrix, not necessarily stochastic.

The language recognized by a NFA is the set of words for which there exists a path in the computation from the starting state to one of the accepting states, more formally:

$$L_A = \{\omega \in \Sigma^* \mid \varphi M(\omega) \eta \geq 1\}.$$

2.3 Probabilistic finite state automata

We now consider a third variant of one-way finite automata, where an input symbol can still trigger multiple transitions from a given state, but the device is only able to follow one of those transitions, which is chosen randomly. Such a machine is called *one way probabilistic finite state automaton* (PFA), and its matrix representation

$$A = \langle \varphi, \{M(\sigma)\}_{\sigma \in \Sigma}, \eta \rangle$$

is such that φ is a stochastic vector called *initial distribution*, where $(\varphi)_i$ is the probability of A starting its computation in the i-th state, and $M(\sigma)$ is a stochastic matrix, where each entry $(M(\sigma))_{i,j}$ represents the probability of transitioning from the i-th state to the j-th state upon reading σ, while η is still the characteristic vector of the accepting states.

The graph representation of a PFA is similar to the one of a NFA, where the edges are also labelled by their corresponding probability (see Figure 2.3).

To each word ω we can associate an *acceptance probability*

$$p_A(\omega) = \varphi M(\omega) \eta,$$

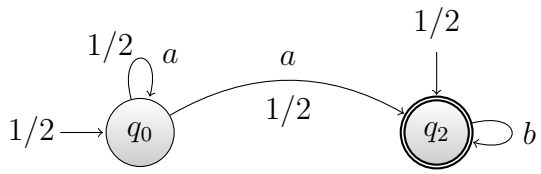


Figure 2.3: PFA over the alphabet $\{a, b\}$. By choosing $k \in \mathbb{N} \setminus \{0\}$ and setting the cut point at $\frac{3}{2^{k+2}}$, the above PFA recognizes the language $L_k = \{a^h b^* \mid h < k\}$. We omit to label the probability of an edge representing a deterministic transition. The labels on the edges indicating the initial states denote the initial distribution.

which represents the probability of A reaching an accepting state by performing a computation on ω. We say that A accepts the language L *with cut point* $\lambda \in [0, 1]$ if

$$L = \{\omega \in \Sigma^* \mid p_A(\omega) > \lambda\},$$

and we say that λ is *isolated by* $\delta \in [0, 1]$ if

$$p_A(\omega) \begin{cases} \geq \lambda + \delta & \text{if } \omega \in L \\ \leq \lambda - \delta & \text{if } \omega \notin L. \end{cases} \tag{2.1}$$

Throughout this thesis, we only consider PFAs with an isolated cut point.

Given a PFA A with an isolated cut point λ, we say that another PFA A' is *equivalent to* A if there exists a value λ', which is isolated for A', such that $L_{A,\lambda} = L_{A',\lambda'}$. Notice that the isolation of the two cut points may be different.

2.4 Automata and regular languages

It is known that both DFAs and NFAs recognize exactly the class of regular language, although the conversion from NFA to DFA, through the well-known subset construction, may require an exponential blow-up in the number of states [50]. Rabin showed that neither adding probabilism with isolated cut point acceptance helps in increasing the computational power. In fact he provided a constructive conversion from PFAs with isolated cut point to DFAs, which requires again an exponential increase of the state complexity [49].

Since all the classical finite state automata models we consider have the same computational power, in this thesis we will only consider regular languages. A regular language is called *reversible* if it is recognized by a DFA where, for each symbol $\sigma \in \Sigma$, the transition function $\tau(\_, \sigma)$ is bijective:

$$\forall \sigma \in \Sigma, \forall q \in Q \;\; \exists! \; \bar{q} \text{ such that } \tau(\bar{q}, \sigma) = q.$$

This means that, for each state of the automaton and for each alphabet symbol, it is always possible to determine the previous state, thus making the computation reversible. This type of DFA is called *reversible automaton*. By using the matrix representation, we can identify reversible automata by the ones where transitions are described by *permutation matrices*, where there is exactly one value set to 1 for each row and for each column.

A language $L \subseteq \Sigma^*$ is *unary* if $|\Sigma| = 1$, i.e. it is defined on an alphabet containing only one symbol. A unary language $L \subseteq \{\sigma\}^*$ is *d-periodic*, for a given integer d, if it holds

$$\sigma^n \in L \Leftrightarrow \sigma^{n+d} \in L,$$

for every integer $n \geq 0$. We say that L is *properly d-periodic* if d is the minimal integer such that L is d-periodic, and we say that L is *periodic* (or *cyclic*) if there exists an integer $d > 0$ such that it is d-periodic. A language L is *ultimately periodic* if there exist two integers $N \geq 0$, $d > 0$ such that $\sigma^n \in L \Leftrightarrow \sigma^{n+d} \in L$, for all $n \geq N$.

We call *unary automaton* an automaton working on a unary alphabet, and we will always assume unary automata to be in canonical form, i.e. $A = \langle \varphi, M, \eta \rangle$, such that the transition matrix M is in the form given in (1.2). By calling m the number of ergodic components, the canonical form induces a partition of the set of states into $m + 1$ classes, according to the transient and ergodic classes defined by the corresponding sets of indices. This implies that the vectors φ and η can be written as $\varphi = \bigoplus_{j=1}^{m+1} \varphi_j$ and $\eta = \bigoplus_{j=1}^{m+1} \eta_j$ according to the same state partition. The states associated to the ergodic components will be called *ergodic states*, while the remaining will be called *transient states*. When associated to automata, the intuitive reading of the matrix 1.2 is the following: the submatrices $M_1, M_2, \ldots, M_m$ describe the transitions within each of the m ergodic classes of states, T_{m+1} describes the transitions within the transient states, while the matrices $T_1, T_2, \ldots, T_m$ describe the transitions from the transient class to each of the m ergodic classes of states.

It is easy to see that a unary DFA is always in the form

$$A = \langle \Sigma = \{\sigma\}, Q, q_0, \tau, F \rangle,$$

such that

- $|Q| = t + p$ for some $t, p \in \mathbb{N}$,
- $Q = \{q_0, \ldots, q_{t-1}, q_t, \ldots, q_{t+p-1}\}$,
- $\tau(q_i, \sigma) = q_{i+1}$ $\forall 0 \leq i \leq t + p - 2$,

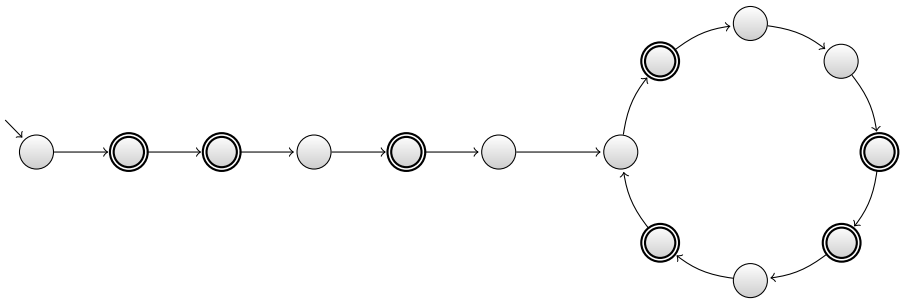


Figure 2.4: Example of unary DFA. When depicting unary automata we will always omit the symbol label, since it would be redundant.

- $\tau(q_{t+p-1}, \sigma) = q_t$.

The graph representation of a generic unary DFA is shown in Figure 2.4. It is easy to see that the first t states $\{q_0, \ldots, q_{t-1}\}$ are the transient component of A (the initial path in Figure 2.4), while the last p states $\{q_t, \ldots, q_{t+p-1}\}$ form the only ergodic component (the final cycle). This implies that *unary regular languages* form *ultimately periodic sets*, as stated by the following

Theorem 2.4.1. *Let $L \subseteq \{\sigma\}^*$ be a unary regular language. Then, there exist two integers $t \geq 0$ and $p > 0$ such that, for any $n \geq t$, we have $\sigma^n \in L$ if and only if $\sigma^{n+p} \in L$.*

A similar property holds for the nondeterministic case: for any NFA recognizing a unary language, there exists an equivalent unary NFA in *Chrobak normal form* [22], i.e. consisting of an initial deterministic path, which ends in a state connected to a set of disjoint deterministic cycles, via nondeterministic transitions. An example of NFA in Chrobak normal form is given in Figure 2.5.

In terms of matrix representation, a NFA in Chrobak normal form can be described as

$$A = \langle \varphi, M, \eta \rangle,$$

where M is in canonical form and can be written as in (1.2), where

- $T_{m+1} \in \{0,1\}^{l_{m+1} \times l_{m+1}}$ describes the initial path of length l_{m+1}, so we have

$$(T_{m+1})_{i,k} = \begin{cases} 1 & \text{if } k = i+1 \\ 0 & \text{otherwise,} \end{cases}$$

- each $M_j \in \{0,1\}^{l_j \times l_j}$ defines a deterministic cycle of length l_j, which means

$$(M_j)_{i,k} = \begin{cases} 1 & \text{if } k \equiv i+1 \pmod{l_j} \\ 0 & \text{otherwise,} \end{cases}$$

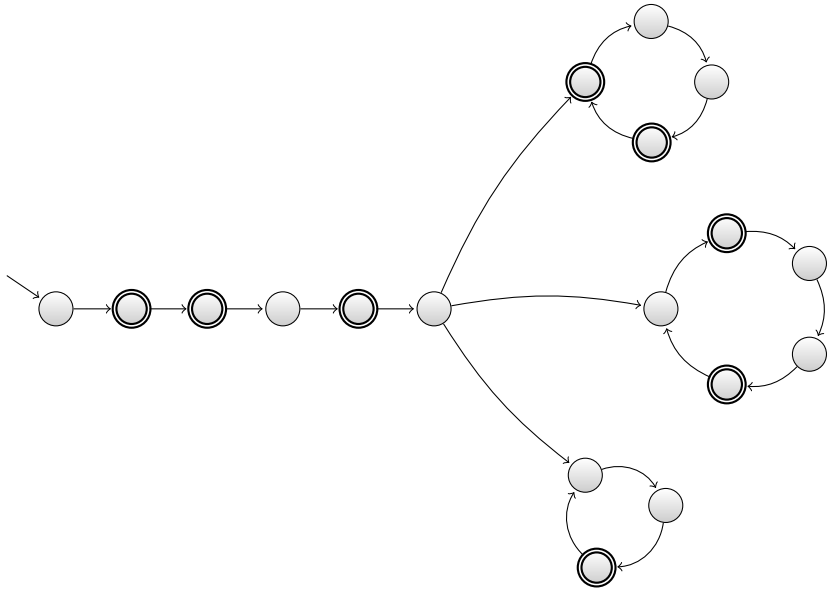


Figure 2.5: Example of NFA in Chrobak normal form with 3 ergodic components.

- for each $1 \leq j \leq m$, $T_j \in \{0,1\}^{l_{m+1} \times l_j}$ describes the nondeterministic connection between the last state of the path and one state of the j-th cycle, more formally

$$(T_j)_{i,k} = \begin{cases} 1 & \text{if } (i,k) = (l_{m+1}, 1) \\ 0 & \text{otherwise.} \end{cases}$$

The initial vector is such that $\varphi_j = 0$ for each $1 \leq j \leq m$ and $\varphi_{m+1} = e_1$, i.e., the only initial state is the first of the transient path, while the final vector η can be any boolean vector, which means that in each cycle and in the initial path there can be more than one accepting state.

It is shown in [22] that each unary n-state NFA can be turned into an equivalent NFA in Chrobak normal form with at most n states in the cycles and $O(n^2)$ states in the initial path (see [26] for a finer estimation).

In Chrobak normal form, the only nondeterministic decision is taken in the last state of the initial path, by choosing one of the transitions leading to the cycles. Notice that, for each cycle, there is exactly one transition from the last state of the path to one state in the cycle; such a state can be considered the "initial state of the cycle".

We quickly recall some complexity results on finite automata and cyclic languages. For properly d-cyclic languages, with d factorizing as $d = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n}$, we have the following optimal state complexities for acceptance:

- minimal DFAs have d states,

- let m and t be the number of states of minimal NFAs and PFAs, respectively. Then, from [35, 41] we have that

$$\sum_{i=1}^{n} p_i^{k_i} \leq t \leq m \leq d.$$

  Clearly, these upper and lower limits coincide whenever d is a prime power.

Notice that, for cyclic languages, it is always possible to have a minimal unary NFA in Chrobak normal form, which is made of one or more deterministic disjoint cycles. Moreover, the following two properties will be useful in the study of cyclic languages:

Lemma 2.4.2 ([22])**.** *Let L be a cyclic language accepted by a* NFA *with m disjoint cycles of length $l_1, \ldots, l_m$, and let $\ell = \mathrm{lcm}\{l_1, \ldots, l_m\}$. Then, L is ℓ-cyclic.*

Lemma 2.4.3 ([43])**.** *Let L be a cyclic language accepted by a* PFA *with m ergodic classes of periods $l_1, \ldots, l_m$, and let $\ell = \mathrm{lcm}\{l_1, \ldots, l_m\}$. Then, L is ℓ-cyclic.*

Chapter 3

A normal form for unary probabilistic automata

A fundamental result by Rabin [49] states that PFAs with isolated cut-points accept only regular languages. (Throughout this thesis, we will consider only isolated cut-points.)

In [43], the authors observed that for each integer N there exists a language accepted by a unary 2-state PFA such that any equivalent DFA requires at least N states. This means that it is impossible to get an upper bound on the number of states needed to simulate unary PFAs by equivalent DFAs or NFAs. Actually, this unbounded number of states is in the initial path of the resulting DFA, i.e., it is related to the "nonperiodic" part of the language. For the cyclic part the situation is different: in the same paper the authors proved that each unary n-state PFA can be simulated by a DFA with $e^{O(\sqrt{n \cdot \ln n})}$ states in its cycle. This is exactly the tight upper bound for the number of the states of a DFA simulating a given unary n-state NFA obtained in [22]. This result stimulated further investigations, with the aim of comparing the sizes of unary DFAs, NFAs, and PFAs, in particular considering cyclic languages or the periodic parts of regular languages (see, e.g., [41, 27, 14]).

In this Chapter, we give a closer look to the structure of unary PFAs. In particular, we want to investigate the possibility of extending Chrobak normal form to the probabilistic case. In other words, given a unary n-state PFA, we are wondering whether it is possible to get an equivalent PFA, without significantly increasing the number of states, where probabilistic decisions can be taken only in one state and at most one time during each computation. It is not difficult to prove that this cannot be done, due to the cost of the "nonperiodic" part of the language, that can be unbounded. On the other hand, the above mentioned simulation result from [43] suggests the idea of restricting the attention to unary cyclic languages: it is known that for these languages turning a NFA in Chrobak

normal form does not increase the total number of states. In this Chapter, we prove that the same does not hold in the probabilistic case. In fact, we show the existence of a unary cyclic language accepted by a PFA with fewer states than any equivalent PFA in Chrobak normal form (the natural extension of the Chrobak normal form for NFAs). We then propose a different kind of normal form, called *cyclic normal form*. In this form, a PFA is a collection of disjoint deterministic cycles, each containing exactly one final state. During a computation, the only nondeterministic decision is taken at the beginning, to select the initial state according to a probabilistic distribution. Our main result shows that each n-state unary PFA accepting a cyclic language can be converted into a unary PFA in cyclic form *without increasing the number of states*. As a consequence, a PFA in cyclic form can be smaller than any equivalent PFA in Chrobak normal form. In the case of nondeterministic devices, the two forms are closely related (even when dropping the restriction to the cyclic case): each NFA in Chrobak normal form can be easily converted into the nondeterministic counterpart of cyclic form and vice versa, preserving, in both transformations, the number of states.

Finally, we discuss the natural extension of the cyclic normal form to all unary regular languages. The results of this Chapter were published in [18, 19].

3.1 Natural extension of Chrobak normal form

To define a probabilistic version of Chrobak normal form, it seems natural to replace the only possible nondeterministic choice by a probabilistic choice. In other words, from the last state of the initial path, the automaton chooses one of the possible cycles (and thus the corresponding initial state) according to a probabilistic distribution (see Figure 3.1). In the matrix representation, a PFA *in Chrobak normal form* can be described as

$$A = \left\langle \varphi = \bigoplus_{j=1}^{m+1} \varphi_j, \quad M, \quad \eta = \bigoplus_{j=1}^{m+1} \eta_j \right\rangle,$$

where M is in canonical form, having m ergodic components, and can be written as in (1.2), where

- $T_{m+1} \in \{0,1\}^{l_{m+1} \times l_{m+1}}$, as in the nondeterministic case, describes the initial path of length l_{m+1}, so we have

$$(T_{m+1})_{i,k} = \begin{cases} 1 & \text{if } k = i+1 \\ 0 & \text{otherwise,} \end{cases}$$

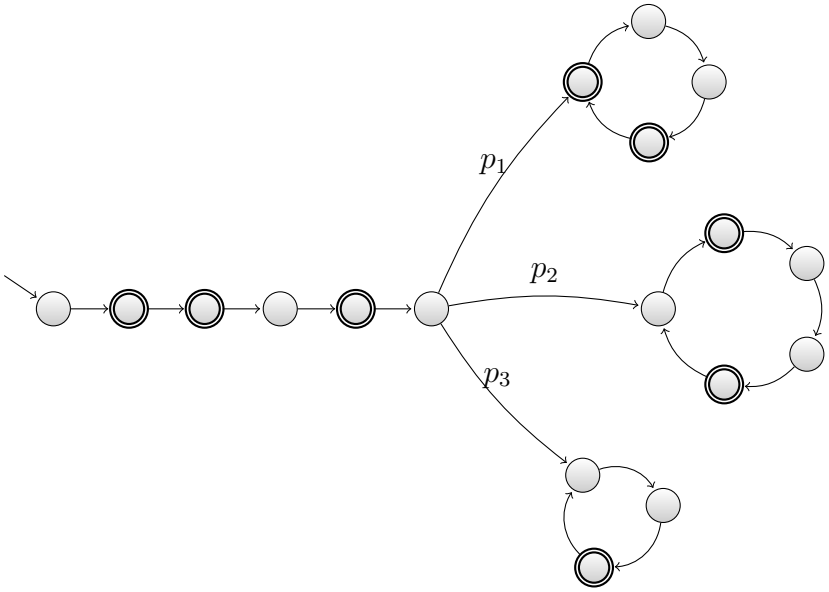


Figure 3.1: Example of PFA in Chrobak normal form, with the constraint $p_1 + p_2 + p_3 = 1$.

- each $M_j \in \{0,1\}^{l_j \times l_j}$, as in the nondeterministic case, defines a deterministic cycle of length l_j, which means

$$(M_j)_{i,k} = \begin{cases} 1 & \text{if } k \equiv i+1 \pmod{l_j} \\ 0 & \text{otherwise,} \end{cases}$$

- for each $1 \leq j \leq m$, $T_j \in \{0,1\}^{l_{m+1} \times l_j}$ describes a probabilistic transition from the last state of the path to one state of the j-th cycle, therefore we need to add the probability constraint

$$(T_j)_{i,k} \begin{cases} \in [0,1] & \text{if } (i,k) = (l_{m+1}, 1) \\ = 0 & \text{otherwise.} \end{cases} \quad \text{and} \quad \sum_{j=1}^{m} (T_j)_{l_{m+1},1} = 1.$$

Again, the initial vector is such that $\varphi_j = 0$ for each $1 \leq j \leq m$ and $\varphi_{m+1} = e_1$, i.e. the initial distribution is concentrated on the first state of the transient component. The final vector η can be any boolean vector, which means that in each cycle there can be more than one accepting state.

We consider the problem of converting a PFA in Chrobak normal form into an equivalent DFA.

Theorem 3.1.1. *For each unary n-state* PFA *M in Chrobak normal form, there exists an equivalent* DFA *M' with $e^{O(\sqrt{n \cdot \ln n})}$ states. Furthermore, the number of states in the initial path of M' is the same as in the initial path of M (besides a possible dead state, if M does not have any cycle).*

Proof. Essentially, it is possible to apply the same construction given in [22] for the conversion from NFAs in Chrobak normal form into DFAs: the initial path of the resulting automaton coincides with that of the given automaton, the cycle simulates "in parallel" all the cycles of the given automaton. To do this, it is enough to take, as length of the cycle, the least common multiple of cycle lengths in the given automaton. The only difference from the nondeterministic case is in the choice of final states in the cycle. Each state of M' represents a tuple of states of M (one for each cycle in the original automaton). In order to make final a state of M', in the nondeterministic case it was enough to have a final state of M in the corresponding tuple. In the probabilistic case, we have to calculate the sum of the probabilities of entering the cycles whose states in the tuple are final. The state of M' under consideration is final if and only if such a sum exceeds the cut-point.

For an upper bound on the number of states, we consider the function

$$F(n) = \max\{\mathrm{lcm}\{x_1, x_2, \ldots, x_k\} \mid x_1 + x_2 + \ldots x_k = n\}.$$

The best known approximation of the above function is due to Szalay [53]:

$$F(n) = e^{O(n \log(n))},$$

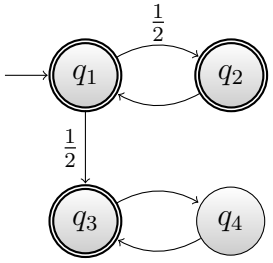
whence the result follows. ∎

We will show now that with this definition we cannot get descriptional complexity properties similar to those for NFAs in Chrobak normal form. In fact, this form does not match neither the bound on the number of states in the initial path (i.e., the noncyclic part of the language) nor that on the number of states in the cycles.

Let us start by considering the initial path. For each integer $k > 0$ we consider the following language

$$L_k = \{\sigma^h | h \leq 2k\} \cup \{\sigma^{2h+1} | h \geq k\}.$$

Clearly L_k can be recognized by the following 4-state PFA



with cut-point $\frac{1}{2^k} - \frac{1}{2^{k+2}}$ isolated by $\frac{1}{2^{k+2}}$. In order to show this, we first notice that the computation on words of odd length may only end in states q_2 and q_3, thus all such words are always accepted, and the computation on words of even length may only end in states q_1 and q_4, so they are accepted only if the automaton is still in the transient component at the end of the computation. This implies that the probability of accepting words $\sigma^h \in L_k$ is either 1 if h is odd, or greater than $\frac{1}{2^k}$, if h is even and not bigger than $2k$ (the worst case is when $h = 2k$). On the other hand, if h is even and at least $2k + 2$, the probability of the computation ending in the accepting state q_1 is not greater than $\frac{1}{2^k} - \frac{1}{2^{k+1}}$ (this lower bound is reached when $h = 2k + 2$). Moreover, it is not difficult to verify that each DFA accepting it must have an initial path of at least $2k$ states. This allows us to state the following:

Theorem 3.1.2. *For every $n \in \mathbb{N}$, there exists a language L_n such that*

- *L_n is accepted by a* PFA *with 4 states,*
- *if a* DFA *made of an initial path of t states and a cycle of p states recognizes L_n, then it must hold $t \geq n$.*

In the light of Theorem 3.1.1, this also implies that each PFA for L_n in Chrobak normal form must have an initial path of at least n states. Hence, we have proved the following:

Corollary 3.1.3. *For all integers n there exists a language L_n which is accepted by a* PFA *with 4 states, but requires more than n states to be accepted by a* PFA *in Chrobak normal form.*

Now, we move our attention to the cyclic part of automata and languages. We will show that, even to express this part, PFAs in Chrobak normal form can be bigger than PFAs. This result will be stated considering cyclic languages.

We recall that, as observed in [35], each NFA in Chrobak normal form accepting a cyclic language can be reduced to one made of a set of disjoint deterministic cycles (namely, the initial path in Figure 2.5 is removed).

We can do the same in the probabilistic case, reducing the Chrobak normal form for PFAs accepting unary languages to a collection of disjoint cycles, each one having *exactly* one initial state. At the beginning of the computation, a probabilistic distribution is used to select one among these possible initial states.

In the following, the next lemma will be useful:

Lemma 3.1.4. *Let L be a properly d-cyclic language accepted by a* PFA *with an isolated cut-point and m ergodic components of periods $l_1, \ldots, l_m$. If, according to*

the Fundamental Theorem of Arithmetic, d factorizes as $p_1^{k_1}p_2^{k_2}\cdots p_n^{k_n}$, then for each $1 \leq i \leq n$ there exists $1 \leq j \leq m$ such that $p_i^{k_i}$ divides l_j.

Proof. By [41, Thm. 2.7], we know that L must be ℓ-cyclic, for $\ell = \mathrm{lcm}(l_1, \ldots l_m)$. Indeed, L being *properly* d-cyclic, d must divide ℓ. Hence, for some integer $\kappa \geq 0$, we have that

$$\kappa d = \ell = \mathrm{lcm}(l_1, \ldots, l_m). \tag{3.1}$$

Now, for a suitable $s \geq n$, let $\prod_{t=1}^{s} p_t^{\gamma_t}$ be the prime factorization of ℓ where, as usual, we let $\gamma_t = \max\{\alpha \in \mathbb{N} \mid p_t^{\alpha} \text{ divides } l_j \text{ for } 1 \leq j \leq m\}$. Equation (3.1) can thus be rewritten as

$$\kappa \prod_{i=1}^{n} p_i^{k_i} = \prod_{t=1}^{s} p_t^{\gamma_t}.$$

This clearly shows that, for each $1 \leq i \leq n$ there must exist $1 \leq t \leq s$ such that ${p_i}^{k_i}$ divides $p_t^{\gamma_t}$. In turn, by definition of least common multiple, we have that $p_t^{\gamma_t}$ must divide l_r, for some $1 \leq r \leq m$, whence the result follows. ∎

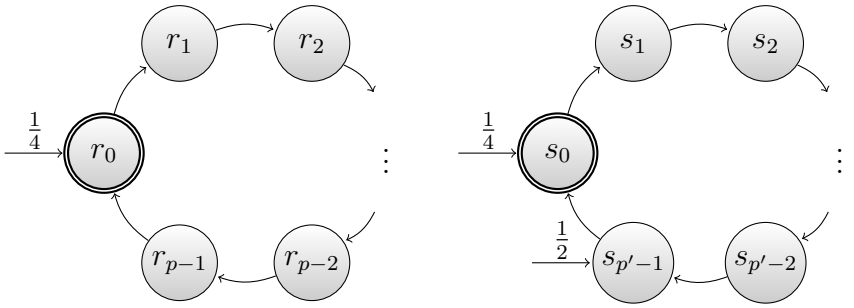
We are now able to prove that there are languages for which PFAs in Chrobak normal form cannot be as small as PFAs:

Theorem 3.1.5. *There exist infinitely many cyclic languages accepted by* PFA*s smaller than each equivalent* PFA *in Chrobak normal form.*

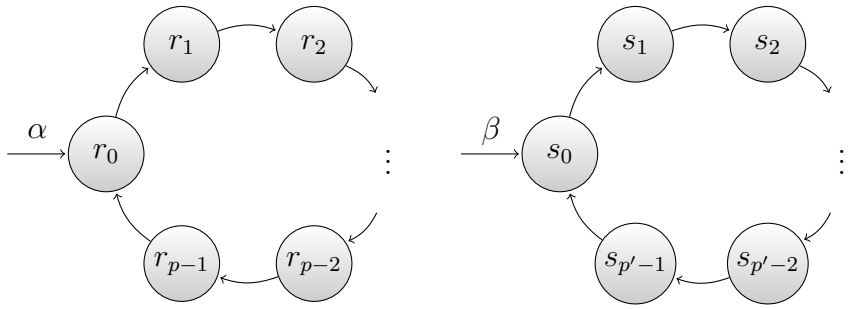
Proof. Given two primes $p < p'$, with $p, p' \geq 2$, consider the language

$$L_{p,p'} = (a^{p \cdot p'})^* + a(a^{p'})^*. \tag{3.2}$$

We can verify that $L_{p,p'}$ is accepted with isolated cut-point $\frac{3}{8}$ by the following PFA M:



By Lemma 3.1.4, this implies that each PFA for $L_{p,p'}$ must have either at least two ergodic components of periods which are multiples of p and p' respectively, or at least one ergodic component of period multiple of $p \cdot p'$. Let A be a PFA in Chrobak normal form for $L_{p,p'}$ with isolated cut point λ. Since $p \cdot p' > p + p'$ for primes $p' > p \geq 2$ we are considering, the only possibility to have A not bigger than M is that A is isomorphic to the following automaton:



for a suitable choice of α, β (i.e., the initial vector φ) and accepting states (i.e., the final vector η). We define variables α_i, β_j, for $i \in \{0, 1, \ldots, p-1\}, j \in \{0, 1, \ldots, p'-1\}$ as follows:

$$\alpha_i = \begin{cases} \alpha & \text{if } r_i \text{ is accepting} \\ 0 & \text{otherwise,} \end{cases} \qquad \beta_j = \begin{cases} \beta & \text{if } s_j \text{ is accepting} \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

The probability of accepting a word a^h on A is exactly $\alpha_{h \bmod p} + \beta_{h \bmod p'}$. In particular, considering the definition of the language, we get the following inequalities:

$$\text{from } \varepsilon \in L_{p,p'}: \qquad \alpha_0 + \beta_0 > \lambda \tag{3.4}$$

$$\text{from } a^p \notin L_{p,p'}: \qquad \alpha_0 + \beta_{p \bmod p'} < \lambda \tag{3.5}$$

$$\text{from } a^{p'} \notin L_{p,p'}: \qquad \alpha_{p' \bmod p} + \beta_0 < \lambda \tag{3.6}$$

$$\text{from } a^{kp'+1} \in L_{p,p'}, \text{ for all } k \in \mathbb{N}: \qquad \alpha_{(kp'+1) \bmod p} + \beta_1 > \lambda. \tag{3.7}$$

From the first three inequalities we get that $\alpha_0 > \alpha_{p' \bmod p}$, and $\beta_0 > \beta_{p \bmod p'}$, therefore both r_0 and s_0 must be accepting, and $\alpha_0 = \alpha$, $\beta_0 = \beta$, while $\alpha_{p' \bmod p} = \beta_{p \bmod p'} = 0$. Because of (3.5) and (3.6), both α_0 and β_0 cannot reach λ on their own, so we have $\alpha, \beta < \lambda$. This implies that in (3.7), for each value of k, neither β_1 nor $\alpha_{(kp'+1) \bmod p}$ can be zero. Since p and p' are coprime, it holds $\{(kp'+1) \bmod p \mid k \in \mathbb{N}\} = \{0, 1, \ldots, p\}$, so all states $r_0, r_1, \ldots, r_{p-1}$ must be accepting, which contradicts $\alpha_{p' \bmod p} = 0$.

∎

3.2 Normal form for unary PFAs accepting cyclic languages

Theorem 3.1.5 shows that the conversion of PFAs into Chrobak normal form requires in general an increase in the number of states. To overcome this problem, we here define a new normal form for PFAs recognizing periodic languages, called

cyclic normal form. We will prove that for any unary PFA accepting a periodic language there exists an equivalent PFA in cyclic normal form with at most the same number of states. As a consequence, each periodic language admits a minimal PFA in cyclic normal form.

Definition A unary PFA $A = \langle \varphi, M, \eta \rangle$ with s states accepting a cyclic language is in *cyclic normal form* if there exist $l_1, \ldots, l_m \in \mathbb{N}$ such that

- $\sum_{j=1}^{m} l_j = s$,
- $\varphi = \bigoplus_{j=1}^{m} \varphi_j$, where $\varphi_j \in \mathbb{R}^{1 \times l_j}$ and $\sum_{i=1}^{s} (\varphi)_i = 1$,
- $M = \bigoplus_{j=1}^{m} M_j$, where $M_j \in \mathbb{R}^{l_j \times l_j}$ and $M_j = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}$,
- $\eta = \bigoplus_{j=1}^{m} \eta_j$, where $\eta_j \in \mathbb{R}^{l_j \times 1}$ and $\eta_j = e_1^T$.

More intuitively, the graph representation of A is a collection of disjoint deterministic cycles of lengths $l_1, \ldots, l_m$, each of them having a unique final state, which is the first state of the cycle, (vectors $\eta_j = e_1^T$) and more than one initial state (vectors φ_j). An example of PFA in cyclic normal form is the automaton M given in the proof of Theorem 3.1.5.

Now, we are able to prove the main result of this Chapter:

Theorem 3.2.1. *For each unary* PFA *A accepting a cyclic language, there exists an equivalent* PFA *A' in cyclic normal form with at most the same number of states.*

Proof. Let L be the language accepted by A, and d an integer such that L is d-cyclic. If $L = \Sigma^*$ or $L = \emptyset$, then it admits a trivial PFA in cyclic normal form, so we prove the theorem for $L \neq \Sigma^*$ and $L \neq \emptyset$.

Suppose that $A = \langle \varphi, M, \eta \rangle$ recognizes the language L with a δ-isolated cutpoint λ. Assume A is in canonical form with m ergodic components of periods $l_1, \ldots, l_m$, so the matrix describing the transitions of A in h steps has the form given in (1.3). The acceptance probability of a word a^h can be seen as the sum of three types of contributions:

- the probability $\varphi_{m+1} T_{m+1}^h \eta_{m+1}$ of A starting in a transient state and remaining in the transient component throughout the whole computation,

- the probability $\varphi_{m+1}\hat{T}_{j,h}\eta_j$ of A starting its computation in a transient state and ending it in the j-th ergodic component,
- the probability $\varphi_j M_j^h \eta_j$ of A both starting and ending its computation in the j-th ergodic component.

More precisely, we have

$$\varphi M^h \eta = \sum_{j=1}^{m} \left(\varphi_j M_j^h \eta_j + \varphi_{m+1} \hat{T}_{j,h} \eta_j \right) + \varphi_{m+1} T_{m+1}^h \eta_{m+1}.$$

Since $\lim_{h\to\infty} \varphi_{m+1} T_{m+1}^h \eta_{m+1} = 0$, by calling $\tilde{p}(j,h) = \varphi_j M_j^h \eta_j + \varphi_{m+1}\hat{T}_{j,h}\eta_j$, we can write

$$\left| \varphi M^h \eta - \sum_{j=1}^{m} \tilde{p}(j,h) \right| < \varepsilon(h) \tag{3.8}$$

for some decreasing function ε which tends to zero. Because of Equation (3.8) and Lemma 1.1.3, we can find a value $H > 0$ such that for each $1 \leq j \leq m$ it holds

$$\left| \varphi M^{Hd} \eta - \sum_{j=1}^{m} \tilde{p}(j, Hd) \right| < \frac{\delta}{4}, \tag{3.9}$$

$$\forall i, k \geq 0 : \quad |\tilde{p}(j, Hd+i) - \tilde{p}(j, Hd + kl_j + i)| < \frac{\delta}{4m}. \tag{3.10}$$

Intuitively, 3.9 means that, for very long inputs, the probability of ending the computation in the transient component is irrelevant, while 3.10 means that, still for long enough inputs, what really matters is the remainder class with respect to the period, rather than the precise length.

Let us now define, for $1 \leq j \leq m$ and $0 \leq r_j \leq l_j$, the probability distribution

$$p(j, r_j) = \frac{\tilde{p}(j, Hd + r_j)}{\sum_{\gamma=1}^{m} \sum_{i=1}^{l_\gamma} \tilde{p}(\gamma, Hd + i)} \tag{3.11}$$

Let $A' = \langle \varphi', M', \eta' \rangle$, where M' and η' have the form described in Definition 3.2 and

$$\varphi' = \bigoplus_{j=1}^{m} \left(p(j,0), p(j,1), \ldots, p(j, l_j - 1) \right).$$

The event induced by A' is

$$\varphi'(M')^h \eta' = \sum_{j=1}^{m} p(j, h \bmod l_j). \tag{3.12}$$

By defining

$$\lambda' = \frac{\lambda}{\sum_{\gamma=1}^{m} \sum_{i=1}^{l_\gamma} \tilde{p}(\gamma, Hd+i)} \quad \text{and} \quad \delta' = \frac{\delta}{2\sum_{\gamma=1}^{m} \sum_{i=1}^{l_\gamma} \tilde{p}(\gamma, Hd+i)},$$

and by applying (3.9), (3.10), (3.11) and (3.12), we get the following implications:

$$\begin{aligned}
a^h \in L &\Rightarrow a^{Hd+h} \in L \\
&\Rightarrow \varphi M^{Hd+h} \eta \geq \lambda + \delta \\
&\Rightarrow \sum_{j=1}^{m} \tilde{p}(j, Hd+h) \geq \lambda + \frac{3\delta}{4} \\
&\Rightarrow \sum_{j=1}^{m} \tilde{p}(j, Hd+h \bmod l_j) \geq \lambda + \frac{\delta}{2} \\
&\Rightarrow \sum_{\gamma=1}^{m} \sum_{i=1}^{l_\gamma} \tilde{p}(\gamma, Hd+i) \sum_{j=1}^{m} p(j, h \bmod l_j) \geq \lambda + \frac{\delta}{2} \\
&\Rightarrow \sum_{j=1}^{m} p(j, h \bmod l_j) \geq \lambda' + \delta' \\
&\Rightarrow \varphi'(M')^h \eta' \geq \lambda' + \delta', \qquad (3.13)
\end{aligned}$$

and:

$$\begin{aligned}
a^h \notin L &\Rightarrow a^{Hd+h} \notin L \\
&\Rightarrow \varphi M^{Hd+h} \eta \leq \lambda - \delta \\
&\Rightarrow \sum_{j=1}^{m} \tilde{p}(j, Hd+h) \leq \lambda - \frac{3\delta}{4} \\
&\Rightarrow \sum_{j=1}^{m} \tilde{p}(j, Hd+h \bmod l_j) \leq \lambda - \frac{\delta}{2} \\
&\Rightarrow \sum_{\gamma=1}^{m} \sum_{i=1}^{l_\gamma} \tilde{p}(\gamma, Hd+i) \sum_{j=1}^{m} p(j, h \bmod l_j) \leq \lambda - \frac{\delta}{2} \\
&\Rightarrow \sum_{j=1}^{m} p(j, h \bmod l_j) \leq \lambda' - \delta' \\
&\Rightarrow \varphi'(M')^h \eta' \leq \lambda' - \delta'. \qquad (3.14)
\end{aligned}$$

Equations (3.13) and (3.14), together with the fact that L is neither empty nor the whole Σ^*, imply

$$\lambda' + \delta' \leq 1 \quad \text{and} \quad \lambda' - \delta' \geq 0,$$

therefore A' recognizes L with cut-point λ' isolated by δ'. ∎

Notice that, by exploiting the structure of the cyclic normal form, we can determine an upper bound (other than the trivial minimum DFA) on the size of the minimal PFA in Chrobak normal form: in fact, to convert any PFA from

cyclic to Chrobak form, it is sufficient to replace each cycle with initial states $q_1, q_2, \ldots, q_k$, with k copies of that cycle, where the i-th copy has as unique initial state the copy of q_i.

Finally, we remark that a cyclic normal form can be defined also on NFAs by requiring φ to be a boolean vector. By allowing multiple initial states, as we did for the Chrobak normal form for NFAs accepting cyclic languages, the conversion into cyclic normal form does not increase the number of states.

3.3 Normal form for all unary PFAs

We now extend the definition of cyclic normal form in order to accept also non periodic regular languages. This is done by adding an initial path of states.

In this way, a NFA in cyclic normal form is similar to one in Chrobak normal form: there is an initial deterministic path and a set of disjoint deterministic cycles. The differences concern the final states and the nondeterministic transitions: each cycle must contain exactly one final state. However, from the last state in the initial path, many different states, even belonging to the same loop, can be reached in a nondeterministic way. Even in this form, the only nondeterministic choice is taken in the last state of the initial path.

An easy construction can be used to transform each NFA in Chrobak normal form into an equivalent NFA in cyclic normal form and viceversa, by keeping the same initial path and the same set of cycles. Hence, for NFAs these two forms can be considered equivalent, even in terms of the number of states (an example of NFA converted from Chrobak into cyclic normal form is given in Figure 3.2).

The probabilistic version of cyclic normal form is defined by replacing the only nondeterministic choice by a probabilistic distribution, on *all* the states in the cycles, namely, from the last state of the initial path, the automaton chooses one ergodic state, according to such a distribution. In the matrix representation, matrices M_j and T_{m+1} are as in Chrobak normal form, while for matrices T_j, $j = 1, \ldots, m$, the following conditions must be satisfied:

$$(T_j)_{i,k} \begin{cases} \in [0,1] & \text{if } i = t_{m+1} \\ = 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \sum_{j=1}^{m} \sum_{k=1}^{t_j} (T_j)_{t_{m+1},k} = 1.$$

Furthermore, the vector of final states η can be decomposed in $m+1$ vectors $\eta_1, \eta_2, \ldots, \eta_m, \eta_{m+1}$, where, for $i = 1, \ldots, m$, the vector η_i corresponding to the i-th cycle has exactly one component equal to 1.

Refining the argument used in the proof, we can adapt Theorem 3.1.1 to the conversion of PFAs in cyclic normal form into DFAs:

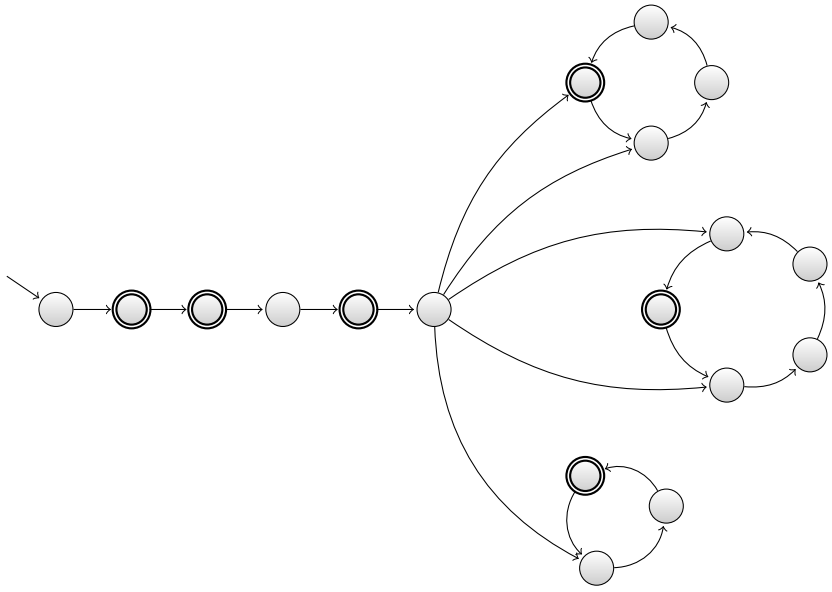


Figure 3.2: NFA in cyclic normal form equivalent to the one in Chrobak normal form of Figure 2.5.

Theorem 3.3.1. *For each unary n-state* PFA M *in cyclic normal form, there exists an equivalent* DFA M' *with $e^{O(\sqrt{n \cdot \ln n})}$ states. Furthermore, the number of states in the initial path of M' is the same as in the initial path of M (besides a possible dead state, if M does not have any cycle).*

Finally, we analyze the cost of converting a PFA into cyclic normal form: Theorems 3.3.1 and 3.1.2 imply that this transformation generates an arbitrarily long initial path, like in the Chrobak normal form. However, Theorem 3.2.1 guarantees that the number of states involved in the cyclic part of the language is not increased after the conversion. To summarize, in the non periodic case Theorems 3.1.5 and 3.2.1 generalize to the following

Corollary 3.3.2. *There exist infinitely many unary regular languages recognized by* PFA*s with less ergodic states than the ergodic states of any equivalent* PFA *in Chrobak normal form.*

Corollary 3.3.3. *For each unary language L recognized by a minimal* PFA *A, there exists a* PFA *A' in cyclic normal form recognizing L with as many ergodic states as A.*

Chapter 4

Descriptional complexity of unary classical automata

In order to get deeper insights into the descriptional power of different paradigms, finite state automata are often studied on very specific tasks, such as recognizing unary or cyclic languages. Periodicity represents a frequent property upon which to test machine descriptional power (see, e.g., [35, 42]). Clearly, to accept properly d-cyclic languages by DFAs, d states are necessary and sufficient. In [41], it is proved that any properly d-cyclic language, where d factorized as product of different prime powers writes as $\prod_{i=1}^{n} p_i^{k_i}$, cannot be accepted by PFAs with less than $\sum_{i=1}^{n} p_i^{k_i}$ states. The same lower bound is obtained in [35] for NFAs. It can also be easily observed that each unary NFA accepting a cyclic language can be converted into an equivalent PFA without increasing the number of states. So, NFAs accepting cyclic languages cannot be more succinct than PFAs. In [41], for any $d = \prod_{i=1}^{n} p_i^{k_i}$, an example is exhibited of properly d-cyclic language accepted by a PFA matching the state lower bound $s = \sum_{i=1}^{n} p_i^{k_i}$, and such that each NFA requires d states, exactly as many as the smallest DFA. On the other hand, for any d it is also possible to give examples of languages for which the size of PFAs and NFAs is the same, matching the lower bound s [42, 41]. As a trivial consequence of this state lower bound, whenever d is a prime power both probabilism and nondeterminism turn out to be unhelpful for state reduction.

In this Chapter, we further deepen these investigations with the aim of comparing, from a descriptional point of view, the use of probabilism and nondeterminism on finite state automata. We prove that *for any d, there exist d-cyclic languages for which d states are necessary (and obviously sufficient) on both* PFA*s and* NFA*s*. In other words, for these languages both probabilism and nondeterminism do not help with respect to determinism. This extends the above mentioned result from the "trivial case" of prime powers to *all integers d*. Next, for infinitely

many d, we exhibit d-cyclic languages requiring t states on PFAs, and m states on NFAs, where $s < t < m < d$. Hence, such language witness that probabilism can help reducing state complexity with respect to nondeterminism, even when neither of them matches either of the two bounds s and d. The results of this Chapter were published in [14, 15]

4.1 A hard language for PFAs

In [41], the properly d-cyclic language $L_d = \{\sigma^{kd} \mid k \in \mathbb{N}\}$ is shown to maximize the gap between the state complexity of PFAs and NFAs. In fact, for $d = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n}$, it is proved that PFAs for L_d reach the theoretical state lower bound $\sum_{i=1}^{n} p_i^{k_i}$, while NFAs must use d states (like DFAs). Here, we consider a family of cyclic languages which are hard, from a descriptional complexity point of view, for both PFAs and NFAs.

Given $d = \prod_{i=1}^{n} p_i^{k_i}$, we define the properly d-cyclic language

$$\mathbf{L} = \{\sigma^h \mid |\{i \mid p_i^{k_i} \text{ divides } h\}| \text{ is even}\}.$$

This language is presented in [38], where it is shown that its minimal NFA is a simple cycle of length d, i.e., it coincides with the minimal DFA. We are going to prove that *d states are necessary and sufficient even on PFAs*. So, this is a case where probabilism does not yield smaller devices.

Theorem 4.1.1. *The language* $\mathbf{L}$ *cannot be recognized with isolated cut point by a* PFA *having less than* $\prod_{i=1}^{n} p_i^{k_i}$ *states.*

Proof. Let $A = \langle \varphi, M, \eta \rangle$ be a PFA in cyclic normal form recognizing $\mathbf{L}$ with δ-isolated cut point λ, and assume by contradiction that the number of states of A is $s < d$. By denoting with m the number of cycles of A, and with $l_1, l_2, \ldots, l_m$ the length of those cycles, we let $\varphi = \bigoplus_{j=1}^{m+1} \varphi_j$, $M = \bigoplus_{j=1}^{m+1} M_j$ and $\eta = \bigoplus_{j=1}^{m+1} \eta_j$ be the state partition induced by the cyclic form, so that the probability of accepting the word σ^h is

$$\varphi M^h \eta = \sum_{j=1}^{m} \left(\varphi_j M_j^h \eta_j\right).$$

For the sake of readability, we let

$$H = \prod_{i=1}^{n} p_i^{k_i - 1} = \frac{d}{\prod_{i=1}^{n} p_i}$$

and, for each $1 \leq i \leq n$, we define the value

$$q_i = \prod_{\substack{t=1 \\ t \neq i}}^{n} p_t^{k_t} \cdot \operatorname{lcm}\{l_j \mid p_i^{k_i} \nmid l_j\}. \tag{4.1}$$

Since $\gcd\{q_i, p_i^{k_i}\} \mid H$, we can find an integer $\alpha_i > 0$ such that

$$p_i^{k_i} \mid H + \alpha_i q_i. \tag{4.2}$$

This property can be derived directly from Theorem 1.0.3, since Equation (4.2) can be seen as $-\alpha_i q_i + \beta_i p_i^{k_i} = H$ for an arbitrary value $\beta_i \in \mathbb{N}$, and we can safely assume that $\alpha_i > 0$, by choosing β_i big enough, since the equation has infinitely many solutions.

Notice that, for each $j \neq i$, it holds

$$p_j^{k_j} \nmid H + \alpha_i q_i, \tag{4.3}$$

since q_i (and thus $\alpha_i q_i$) is a multiple of $p_j^{k_j}$, while H is not.

In order to disprove the hypothesis $s < d$, we are going to consider a limited set of words: the idea is to build such words starting from a string of length H, and considering each string of length $\alpha_i q_i$ as a suffix to be appended to the current word, in order to add to the input length only the divisibility by $p_i^{k_i}$ and leave the divisibility by all other prime powers unchanged.

More formally, we claim that, for every subset of indexes $Z \subseteq \{1, 2, \ldots, n\}$,

$$p_i^{k_i} \mid H + \sum_{z \in Z} \alpha_z q_z \;\Leftrightarrow\; i \in Z.$$

If $Z = \{i\}$ for some $1 \leq i \leq n$, the claim holds because of Equations (4.2) and (4.3). If $|Z| > 1$, without loss of generality we can write $Z = \{1, 2, \ldots, i\}$, for some integer $i > 1$. By induction we assume

$$p_t^{k_t} \mid H + \sum_{z=1}^{i-1} \alpha_z q_z, \quad \forall t < i, \tag{4.4}$$

and

$$p_t^{k_t} \nmid H + \sum_{z=1}^{i-1} \alpha_z q_z, \quad \forall t \geq i. \tag{4.5}$$

Therefore, when we consider the additional suffix of length $\alpha_i q_i$ we obtain

- $p_t^{k_t} \mid H + \sum_{z=1}^{i-1} \alpha_z q_z + \alpha_i q_i, \quad \forall t < i,$
  because of 4.4 and because $p_t^{k_t} \mid \alpha_i q_i$ for each $t < i$,
- $p_i^{k_i} \mid H + \sum_{z=1}^{i-1} \alpha_z q_z + \alpha_i q_i,$
  because of 4.2 and because $p_i^{k_i} \mid \alpha_z q_z$ for each $z < i$,
- $p_t^{k_t} \nmid H + \sum_{z=1}^{i-1} \alpha_z q_z + \alpha_i q_i, \quad \forall t > i,$
  because of 4.3 and because $p_t^{k_t} \mid \alpha_z q_z$ for each $t \neq z$.

In order to prove the theorem, we consider the following set of words:

$$L_H = \left\{\sigma^{H+\sum_{z\in Z}\alpha_z q_z} \;\middle|\; Z \subseteq \{1,\dots,n\}\right\}. \tag{4.6}$$

This set can be partitioned into two languages

$$L_{H^+} = \left\{\sigma^{H+\sum_{z\in Z}\alpha_z q_z} \;\middle|\; Z \subseteq \{1,\dots,n\} \text{ and } |Z| \text{ is even}\right\},$$
$$L_{H^-} = \left\{\sigma^{H+\sum_{z\in Z}\alpha_z q_z} \;\middle|\; Z \subseteq \{1,\dots,n\} \text{ and } |Z| \text{ is odd}\right\}.$$

Clearly, we have $L_{H^+} \subset \mathbf{L}$, $L_{H^-} \subset \mathbf{L}^c$, and it is not hard to see that L_{H^+} and L_{H^-} have exactly the same number of words:

$$|L_{H^+}| = |L_{H^-}| = 2^{n-1}. \tag{4.7}$$

This partitioning implies that, on input σ^n, the PFA A must satisfy the following two properties:

- if $\sigma^h \in L_{H^+}$, then $\sum_{j=1}^m \varphi_j M_j^h \eta_j > \lambda + \delta$,
- if $\sigma^h \in L_{H^-}$, then $\sum_{j=1}^m \varphi_j M_j^h \eta_j < \lambda - \delta$.

The above constraint, together with Equation (4.7), implies

$$\sum_{\sigma^\mu \in L_{H^+}} \sum_{j=1}^m \varphi_j M_j^\mu \eta_j - \sum_{\sigma^\nu \in L_{H^+}} \sum_{j=1}^m \varphi_j M_j^\nu \eta_j > 2^{n-1}(\lambda+\delta) - 2^{n-1}(\lambda-\delta),$$

which can be written as

$$\sum_{j=1}^m \left(\sum_{\sigma^\mu \in L_{H^+}} \varphi_j M_j^\mu \eta_j - \sum_{\sigma^\nu \in L_{H^+}} \varphi_j M_j^\nu \eta_j \right) > 2^n \delta, \tag{4.8}$$

and we are going to show that the left side of the above inequality simplifies to zero. In order to do so, for each $1 \le j \le m$ (i.e., for each cycle of the automaton), we consider the following partition of Z:

$$\Gamma_j = \{i \;\mid\; 1 \le i \le n \text{ and } l_j \nmid \alpha_i q_i\}$$
$$\Lambda_j = \{i \;\mid\; 1 \le i \le n \text{ and } l_j \mid \alpha_i q_i\},$$

and we highlight this partition of Z in the words of L_H as follows:

$$\sigma^{H+\sum_{z\in Z}\alpha_z q_z} = \sigma^{H+\sum_{z\in Z\cap\Gamma_j}\alpha_z q_z + \sum_{z\in Z\cap\Lambda_j}\alpha_z q_z}. \tag{4.9}$$

The reason why we do this is because we can now get rid of the last part of the exponent in Equation (4.9) when considering only the j-th cycle of A: since $\sum_{z\in Z\cap\Lambda_j}\alpha_z q_z$ is a multiple of l_j, reading a string of that length will leave the

current state of A in the j-th cycle unchanged. In other words, the suffix of length $\sum_{z\in Z\cap\Lambda_j}\alpha_z q_z$ is irrelevant for the probability contribution given by the j-th cycle.

More formally, for any $X\subseteq\Gamma_j$ and $Y\subseteq\Lambda_j$, by calling $h(X,Y)=H+\sum_{z\in X}\alpha_z q_z+\sum_{z\in Y}\alpha_z q_z$, we have

$$\langle h(X,Y)\rangle_{l_j}=\langle h(X,\emptyset)\rangle_{l_j},$$

and therefore

$$\varphi_j M_j^{h(X,Y)}\eta_j=\varphi_j M_j^{h(X,\emptyset)}\eta_j.$$

Moreover, the initial assumption $s<d$ implies that A has no cycle whose length is a multiple of all the prime powers which factorize d, therefore no set Λ_j is empty. This allows us to rewrite the left side of 4.8 as follows

$$\sum_{j=1}^{m}\left(\sum_{\sigma^\mu\in L_{H^+}}\varphi_j M_j^\mu\eta_j-\sum_{\sigma^\nu\in L_{H^+}}\varphi_j M_j^\nu\eta_j\right)=$$

$$=\sum_{j=1}^{m}\sum_{X\subseteq\Gamma_j}\left(\sum_{\substack{Y\subseteq\Lambda_j\\ \langle|X\cup Y|\rangle_2=0}}\varphi_j M_j^{h(X,Y)}\eta_j-\sum_{\substack{Y\subseteq\Lambda_j\\ \langle|X\cup Y|\rangle_2=1}}\varphi_j M_j^{h(X,Y)}\eta_j\right)$$

$$=\sum_{j=1}^{m}\sum_{X\subseteq\Gamma_j}\left(\sum_{\substack{Y\subseteq\Lambda_j\\ \langle|X\cup Y|\rangle_2=0}}\varphi_j M_j^{h(X,\emptyset)}\eta_j-\sum_{\substack{Y\subseteq\Lambda_j\\ \langle|X\cup Y|\rangle_2=1}}\varphi_j M_j^{h(X,\emptyset)}\eta_j\right). \tag{4.10}$$

Since $\Lambda_j\neq\emptyset$, we have

$$|\{Y\subseteq\Lambda_j\ \mid\ \langle\,|Y|\,\rangle_2=0\}|=|\{Y\subseteq\Lambda_j\ \mid\ \langle\,|Y|\,\rangle_2=1\}|.$$

Thus, for any given $X\subseteq\Gamma_j$, the words in the set $\left\{\sigma^{h(X,Y)}\ \mid\ Y\subseteq\Lambda_j\right\}$ are equally distributed in L_{H^+} and L_{H^-}. Therefore, the two sums in the brackets of Equation (4.10) have the same number of elements, which are all the same constant value. This implies that the whole sum is equal to zero, thus leading to the contradiction in Equation (4.8)

$$0>2^n\delta.$$

∎

4.2 Probabilism vs nondeterminism

In this section, we prove the existence of languages on which the probabilistic paradigm actually shows its higher descriptional power with respect to nonde-

terminism. Precisely, we design families of cyclic languages for which PFAs are smaller than NFAs which, in turn, are smaller than DFAs.

Given $n, m_1, \ldots, m_n > 0$ and powers $x_{i,j} = p_{i,j}^{k_{i,j}} > 1$ of pairwise different primes, for $1 \leq i \leq n$ and $1 \leq j \leq m_i$, we define

$$\mathbf{L}_{\mathrm{df}} = \left\{ \sigma^h \;\middle|\; \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \langle h \rangle_{x_{i,j}} = 0 \right\}, \quad \text{and} \quad \mathbf{L}_{\mathrm{cf}} = \left\{ \sigma^h \;\middle|\; \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} \langle h \rangle_{x_{i,j}} = 0 \right\}.$$

It is not hard to see that both languages are properly $\prod_{i=1}^{n} \prod_{j=1}^{m_i} x_{i,j}$-cyclic.

Nondeterministic automata for $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$

Let us now show that NFAs for $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$ are always bigger than the equivalent minimal PFAs. We begin by the following technical lemma:

Lemma 4.2.1. *Let L be a properly d-cyclic language, with $d = \prod_{i=1}^{n} p_i^{k_i}$, accepted by a NFA A with m strongly connected components of periods $l_1, \ldots, l_m$. Then, for each $1 \leq i \leq n$, there exists $1 \leq j \leq m$ such that $p_i^{k_i}$ divides l_j.*

Proof. Because of Lemma 2.4.2, we know that L must be ℓ-cyclic, for $\ell = \mathrm{lcm}\{l_1, \ldots l_m\}$. In fact, L being properly d-cyclic, it must be that $d \mid \ell$. Hence, for some integer $\kappa \geq 0$, we have that

$$\kappa d = \ell = \mathrm{lcm}\{l_1, \ldots, l_m\}. \tag{4.11}$$

Now, for a suitable $s \geq n$, let $\prod_{t=1}^{s} q_t^{\gamma_t}$ be the prime factorization of ℓ. Equation (4.11) can thus be rewritten as

$$\kappa \prod_{i=1}^{n} p_i^{k_i} = \prod_{t=1}^{s} q_t^{\gamma_t}.$$

This clearly shows that, for each $1 \leq i \leq n$, there must exist $1 \leq t \leq s$ such that ${p_i}^{k_i} \mid q_t^{\gamma_t}$. In turn, by definition of least common multiple, we have that $q_t^{\gamma_t}$ must divide l_j, for some $1 \leq j \leq m$, whence the result follows. ∎

By considering [35, Thm. 2.1] in connection with Lemma 4.2.1, we get the following result stating a simple normal form for NFAs accepting cyclic languages:

Lemma 4.2.2. (Simple normal form) *Let L be a properly $\prod_{i=1}^{n} p_i^{k_i}$-cyclic language. Then L has a minimal state NFA A in one of the following forms:*

FORM 1 - *A consists of a single deterministic cycle.*

FORM 2 - *A consists of two or more pairwise disjoint deterministic cycles.*

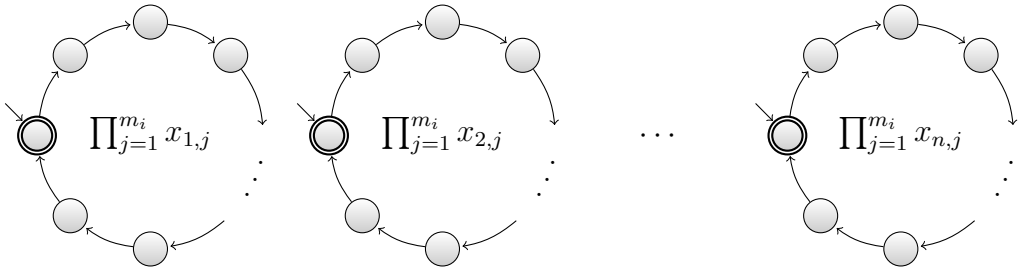


Figure 4.1: Graph representation of the NFA for $\mathbf{L}_{\mathrm{df}}$. The number written in the center of each cycle denotes its length, i.e. the number of states in that cycle.

In both cases, for every $1 \leq i \leq n$, there exists a cycle whose length is a multiple of $p_i^{k_i}$.

We are now ready to give optimal size NFAs for $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$:

Theorem 4.2.3. *The language $\mathbf{L}_{\mathrm{df}}$ defined on prime powers $x_{i,j}$, for $1 \leq i \leq n$ and $1 \leq j \leq m_i$, can be recognized by a minimal NFA with $\sum_{i=1}^{n} \prod_{j=1}^{m_i} x_{i,j}$ states.*

Proof. We start by describing a NFA with $\sum_{i=1}^{n} \prod_{j=1}^{m_i} x_{i,j}$ states recognizing $\mathbf{L}_{\mathrm{df}}$. Our NFA consists of n disjoint simple cycles of length $\prod_{j=1}^{m_i} x_{i,j}$, for $1 \leq i \leq n$. The i-th cycle checks the condition $\bigwedge_{j=1}^{m_i} \langle h \rangle_{x_{i,j}} = 0$ on input σ^h by using only one initial state which is also the unique final state. Such an automaton is shown in Figure 4.1.

The correctness of this automaton is trivial, so we are going to prove its minimality. For the sake of readability, let $X_i = \prod_{j=1}^{m_j} x_{i,j}$ for every $1 \leq i \leq n$, and fix $1 \leq \xi \leq n$. Clearly, $\sigma^{X_\xi} \in \mathbf{L}_{\mathrm{df}}$. We show that any minimal NFA A for $\mathbf{L}_{\mathrm{df}}$ must have at least a cycle of length multiple of X_ξ. We can assume that A is in simple normal form. Suppose that A is made of a single cycle: since $\mathbf{L}_{\mathrm{cf}}$ is properly $\prod_{i=1}^{n} \prod_{j=1}^{m_i} x_{i,j}$-cyclic, we have, by Lemma 4.2.2, that the only cycle has a length greater than the number of states given in the statement. Now, let us assume that A has FORM 2. Suppose that A accepts σ^{X_ξ} on a cycle of length $\ell > 0$. By letting $X = (\prod_{i=1}^{n} X_i)/X_\xi$, we have that the string $\sigma^{X_\xi + X\ell}$ is also accepted by A. Hence, $\sigma^{X_\xi + X\ell}$ belongs to $\mathbf{L}_{\mathrm{df}}$ as well, so there must exists $1 \leq s \leq n$ such that

$$X_\xi + X\ell = \alpha X_s, \tag{4.12}$$

for some positive integer α. Therefore we have that $s = \xi$. In fact, assume by contradiction that $s \neq \xi$ and rewrite Equation (4.12) as $X_\xi = \alpha X_s - X\ell$. Such a Diophantine equation, with α and ℓ as indeterminates, admits solutions if and only if $\gcd(X_s, X)$ divides X_ξ. If we assume $s \neq \xi$, then we have $\gcd(X_s, X) = X_s$, which divides X_ξ if and only if $s = \xi$, leading to a contradiction. So, by having

$s = \xi$, Equation (4.12) becomes $X\ell = (\alpha - 1)X_\xi$, and since X_ξ does not divide X, we must conclude that ℓ is multiple of X_ξ.

By iterating this reasoning, we obtain that for every $1 \leq i \leq n$ a cycle must exist in A whose length is multiple of X_i. The "most succinct way" to guarantee this property is that A contains n disjoint cycles whose lengths are X_i, for every $1 \leq i \leq n$. This directly implies that the state lower bound for NFAs accepting $\mathbf{L}_{\mathrm{df}}$ is $\sum_{i=1}^{n} \prod_{j=1}^{m_i} x_{i,j}$. ∎

Let us now study minimal NFAs for languages in the form $\mathbf{L}_{\mathrm{cf}}$. By standard properties of boolean operators, one may easily turn a language in the form $\mathbf{L}_{\mathrm{cf}}$ into a language defined by a disjunction of conjunctions. At this point, an NFA for $\mathbf{L}_{\mathrm{cf}}$ can be obtained in a way similar to the NFA for $\mathbf{L}_{\mathrm{df}}$ given in Theorem 4.2.3. We are going to show that the size of the NFA resulting from this approach is the best possible.

Theorem 4.2.4. *The language $\mathbf{L}_{\mathrm{cf}}$ defined on prime powers $x_{i,j}$, for $1 \leq i \leq n$ and $1 \leq j \leq m_i$, can be recognized by a minimal* NFA *with $\prod_{i=1}^{n} \sum_{j=1}^{m_i} x_{i,j}$ states.*

Proof. Our first step is to transform the modularity condition for $\mathbf{L}_{\mathrm{cf}}$ into an equivalent disjunction of conjunctions, so we can build a NFA with the same construction as the one used in the proof of Theorem 4.2.3. To this aim, we define a function $\phi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $\phi(t, i) = j$, if $x_{i,j}$ is the i-th prime power of the t-th conjunction in the transformed formula. Formally, ϕ is such that:

- for any $1 \leq t \leq \prod_{i=1}^{n} m_i$ and $1 \leq i \leq n$, we have $\phi(t, i) \in \{1, 2, \ldots, m_i\}$,
- for every $t \neq t'$ there exists an i such that $\phi(t, i) \neq \phi(t', i)$.

By this definition, we get that the n prime powers involved in each of the $\prod_{i=1}^{n} m_i$ conjunctions are exactly the set $\bigcup_{i=1}^{n} \{x_{i,\phi(t,i)}\}$, for some $1 \leq t \leq \prod_{i=1}^{n} m_i$.

The NFA for $\mathbf{L}_{\mathrm{cf}}$, obtained by applying the construction of Theorem 4.2.3, consists of $\prod_{i=1}^{n} m_i$ disjoint simple cycles of length $X_t = \prod_{i=1}^{n} x_{i,\phi(t,i)}$. Indeed, it is not hard to verify that the sum of all those X_t's is exactly $\prod_{i=1}^{n} \sum_{j=1}^{m_i} x_{i,j}$. For each $1 \leq t \leq \prod_{i=1}^{n} m_i$, the cycle of length X_t checks the condition $\bigwedge_{i=1}^{n} \langle h \rangle_{x_{i,\phi(t,i)}} = 0$ on input σ^h by using only one initial state which is also the unique final state.

In order to prove the minimality of our automaton, we show that every NFA A for $\mathbf{L}_{\mathrm{cf}}$ must have a cycle whose length is a multiple of X_t, for each $1 \leq t \leq \prod_{i=1}^{n} m_i$. As in the proof of Theorem 4.2.3, let us assume A to be in simple normal form. Consider the input σ^{X_t} which is clearly in $\mathbf{L}_{\mathrm{cf}}$. Suppose A accepts σ^{X_t} on a cycle of length $\ell > 0$ which, by contradiction, is not a multiple of X_t.

This implies that there must exist a $1 \leq r \leq n$ satisfying $x_{r,\phi(t,r)} \nmid \ell$. Now, by letting

$$X = \frac{\prod_{j=1}^{m_r} x_{r,j}}{x_{r,\phi(t,r)}},$$

the word $\sigma^{X_t + X\ell}$ is still accepted by A and hence belongs to $\mathbf{L}_{\mathrm{cf}}$. However, for $1 \leq j \leq m_r$:

- if $j = \phi(t,r)$, it holds $x_{r,j} \mid X_t$ and $x_{r,j} \nmid X\ell$, while
- if $j \neq \phi(t,r)$, it holds $x_{r,j} \nmid X_t$ and $x_{r,j} \mid X\ell$.

This shows that, for every $1 \leq j \leq m_r$, we have $x_{r,j} \nmid X_t + X\ell$ implying the contradiction $\sigma^{X_t + X\ell} \notin \mathbf{L}_{\mathrm{cf}}$.

By iterating this reasoning, we obtain that, for every $1 \leq t \leq \prod_{i=1}^{n} m_i$, there must exist in A a cycle whose length is multiple of X_t. The "most succinct way" to guarantee this property is that A contains $\prod_{i=1}^{n} m_i$ disjoint cycles whose lengths are X_t, for every $1 \leq t \leq \prod_{i=1}^{n} m_i$. ∎

Probabilistic automata for $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$

We now give a construction of PFAs for $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$, showing how the probabilistic paradigm can help in saving states with respect to nondeterminism. However, we also show that, except for trivial definitions of $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$ (which will be analyzed in Section 4.2.1), PFAs cannot reach the state lower bound given in [41].

We start by this latter result, showing that, in general, in order to recognize the languages $\mathbf{L}_{\mathrm{df}}$ or $\mathbf{L}_{\mathrm{cf}}$, PFAs with isolated cut point require a size which is strictly greater than the state lower bound.

Theorem 4.2.5. *Let L_1 be a language in the form $\mathbf{L}_{\mathrm{df}}$ and L_2 a language in the form $\mathbf{L}_{\mathrm{cf}}$, both defined on prime powers $x_{i,j}$, for $1 \leq i \leq n$ and $1 \leq j \leq m_i$. If $n \geq 2$ and there exist $1 \leq i_1 \neq i_2 \leq n$ such that $m_{i_1} \geq 2$ and $m_{i_2} \geq 2$, then any minimal* PFA *recognizing with isolated cut point either the language L_1 or L_2 requires more than $\sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{i,j}$ states.*

Note: Any $\mathbf{L}_{\mathrm{df}}$ or $\mathbf{L}_{\mathrm{cf}}$ language having the form stated in this theorem will be called *nontrivial*.

Proof. Without loss of generality we can assume $i_1 = 1$ and $i_2 = 2$. We consider only minimal PFAs in cyclic normal form, thus, if the Theorem does not hold, from Lemma 3.1.4 we have that a minimal PFA recognizing either L_1 or L_2 with $\sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{i,j}$ states must have a transition matrix which is a direct sum of $\sum_{i=1}^{n} m_i$ submatrices of both period and dimension $l_{i,j} = x_{i,j}$, for $1 \leq i \leq n$ and

$1 \leq j \leq m_i$. Therefore, we can assume that both minimal PFAs for L_1 or L_2 have the cyclic normal form

$$\left(\bigoplus_{i=1}^{n} \bigoplus_{j=1}^{m_i} \varphi_{i,j}, \bigoplus_{i=1}^{n} \bigoplus_{j=1}^{m_i} M_{i,j}, \bigoplus_{i=1}^{n} \bigoplus_{j=1}^{m_i} \eta_{i,j} \right), \tag{4.13}$$

where, for every $h \geq 0$, we have $M_{i,j}^h = M_{i,j}^{\langle h \rangle_{x_{i,j}}}$. Hence, the accepting probability of such automaton on input σ^h is:

$$p(\sigma^h) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \varphi_{i,j} M_{i,j}^h \eta_{i,j} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} P(i,j,\langle h \rangle_{x_{i,j}}), \tag{4.14}$$

where we let $P(i,j,\langle h \rangle_{x_{i,j}}) = \varphi_{i,j} M_{i,j}^{\langle h \rangle_{x_{i,j}}} \eta_{i,j}$.

We start by proving by contradiction that *any* PFA A_1 *accepting* L_1 *with isolated cut point* λ *must have more than* $\sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{i,j}$ *states.* As a contradiction, we let A_1 be in the form (4.13), so that the acceptance probability is given by (4.14). For every $1 \leq i \leq n$ and $1 \leq j \leq m_i$, we fix a value $0 < r_{i,j} < x_{i,j}$. Since all $x_{i,j}$'s are pairwise coprime, by Theorem 1.0.1, there exist natural numbers h_1, h_2, h_3, h_4 such that:

- for all $1 \leq j \leq m_1$ it holds $\langle h_1 \rangle_{x_{1,j}} = 0$, and
  for all $2 \leq i \leq n$, $1 \leq j \leq m_i$ it holds $\langle h_1 \rangle_{x_{i,j}} = r_{i,j}$,
- for all $1 \leq j \leq m_2$ it holds $\langle h_2 \rangle_{x_{2,j}} = 0$, and
  for all $1 \leq i \neq 2 \leq n$, $1 \leq j \leq m_i$ it holds $\langle h_2 \rangle_{x_{i,j}} = r_{i,j}$,
- $\langle h_3 \rangle_{x_{1,1}} = 0$,
  for all $2 \leq j \leq m_1$ it holds $\langle h_3 \rangle_{x_{1,j}} = r_{1,j}$,
  $\langle h_3 \rangle_{x_{2,1}} = 0$,
  for all $2 \leq j \leq m_2$ it holds $\langle h_3 \rangle_{x_{2,j}} = r_{2,j}$, and
  for all $3 \leq i \leq n$, $1 \leq j \leq m_i$ it holds $\langle h_3 \rangle_{x_{i,j}} = r_{i,j}$,
- $\langle h_4 \rangle_{x_{1,1}} = r_{1,1}$,
  for all $2 \leq j \leq m_1$ it holds $\langle h_4 \rangle_{x_{1,j}} = 0$,
  $\langle h_4 \rangle_{x_{2,1}} = r_{2,1}$,
  for all $2 \leq j \leq m_2$ it holds $\langle h_4 \rangle_{x_{2,j}} = 0$, and
  for all $3 \leq i \leq n$, $1 \leq j \leq m_i$ it holds $\langle h_4 \rangle_{x_{i,j}} = r_{i,j}$.

Less formally, h_1 (h_2, respectively) denotes the length of a word satisfying only the first (second, respectively) clause, h_3 is a length satisfying only the first literal of the first two clauses, while a word of length h_4 has a behavior on the first two clauses which is complementary to h_3, since it satisfies all literals except the first

ones. Clearly, both σ^{h_1} and σ^{h_2} belong to L_1, while both σ^{h_3} and σ^{h_4} do not belong to L_1. Hence, for (4.14), we have:

- $p(\sigma^{h_1}) = \sum_{j=1}^{m_1} P(1,j,0) + \sum_{i=2}^{n} \sum_{j=1}^{m_i} P(i,j,r_{i,j}) > \lambda$,
- $p(\sigma^{h_2}) = \sum_{j=1}^{m_2} P(2,j,0) + \sum_{\substack{i=1 \\ i \neq 2}}^{n} \sum_{j=1}^{m_i} P(i,j,r_{i,j}) > \lambda$,
- $p(\sigma^{h_3}) = P(1,1,0) + \sum_{j=2}^{m_1} P(1,j,r_{1,j}) + P(2,1,0) + \sum_{j=2}^{m_2} P(2,j,r_{2,j}) + \sum_{i=3}^{n} \sum_{j=1}^{m_i} P(i,j,r_{i,j}) < \lambda$,
- $p(\sigma^{h_4}) = P(1,1,r_{1,1}) + \sum_{j=2}^{m_1} P(1,j,0) + P(2,1,r_{2,1}) + \sum_{j=2}^{m_2} P(2,j,0) + \sum_{i=3}^{n} \sum_{j=1}^{m_i} P(i,j,r_{i,j}) < \lambda$.

Then, we get $p(\sigma^{h_1}) + p(\sigma^{h_2}) > 2\lambda$ and $p(\sigma^{h_3}) + p(\sigma^{h_4}) < 2\lambda$, but $p(\sigma^{h_1}) + p(\sigma^{h_2}) = p(\sigma^{h_3}) + p(\sigma^{h_4})$, a contradiction.

Now, we prove that *any minimal* PFA A_2 *for* L_2 *with isolated cut point* λ *must have more than* $\sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{i,j}$ *states*. Similarly to the previous case, we let A_2 be in the form (4.13). Again, for every $1 \leq i \leq n$ and $1 \leq j \leq m_i$, we fix a value $0 < r_{i,j} < x_{i,j}$ and, by the Chinese Remainder Theorem, there exist natural numbers z_1, z_2, z_3, z_4 such that:

- for all $1 \leq i \leq n$ and $2 \leq j \leq m_i$ it holds $\langle z_1 \rangle_{x_{i,1}} = 0$ and $\langle z_1 \rangle_{x_{i,j}} = r_{i,j}$,
- for all $1 \leq i \leq 2$ and $1 \leq j \neq 2 \leq m_i$ it holds $\langle z_2 \rangle_{x_{i,2}} = 0$ and $\langle z_2 \rangle_{x_{i,j}} = r_{i,j}$, while for all $3 \leq i \leq n$ and $2 \leq j \leq m_i$ it holds $\langle z_2 \rangle_{x_{i,1}} = 0$ and $\langle z_2 \rangle_{x_{i,j}} = r_{i,j}$,
- $\langle z_3 \rangle_{x_{1,1}} = \langle z_3 \rangle_{x_{1,2}} = 0$ and for all $3 \leq j \leq m_1$ it holds $\langle z_3 \rangle_{x_{1,j}} = r_{1,j}$, for all $1 \leq j \leq m_2$ it holds $\langle z_3 \rangle_{x_{2,j}} = r_{2,j}$, and for all $3 \leq i \leq n$ and $2 \leq j \leq m_i$ it holds $\langle z_3 \rangle_{x_{i,1}} = 0$ and $\langle z_3 \rangle_{x_{i,j}} = r_{i,j}$,
- $\langle z_4 \rangle_{x_{2,1}} = \langle z_4 \rangle_{x_{2,2}} = 0$, for all $3 \leq j \leq m_2$ it holds $\langle z_4 \rangle_{x_{2,j}} = r_{2,j}$, for all $1 \leq j \leq m_1$ it holds $\langle z_4 \rangle_{x_{1,j}} = r_{1,j}$, and for all $3 \leq i \leq n$ and $2 \leq j \leq m_i$ it holds $\langle z_4 \rangle_{x_{i,1}} = 0$ and $\langle z_4 \rangle_{x_{i,j}} = r_{i,j}$.

In other words, all z_1, z_2, z_3, z_4 denote lengths of words which, from the third clause on, only satisfy the first literal. On the first two clauses, z_1 (z_2, respectively) satisfies the first (second, respectively) literal of each clause, therefore verifying the language condition, while z_3 (z_4, respectively) satisfies the first two literals of the first (second, respectively) clause, thus leaving one clause not satisfied. Clearly, both σ^{z_1} and σ^{z_2} belong to L_2, while both σ^{z_3} and σ^{z_4} do not belong to

L_2. Thus, we compute the acceptance probabilities according to (4.14) and, by calling

$$\hat{P} = \sum_{i=1}^{2}\sum_{j=3}^{m_i} P(i,j,r_{i,j}) + \sum_{i=3}^{n}\left(P(i,1,0) + \sum_{j=2}^{m_i} P(i,j,r_{i,j})\right),$$

we have

- $p(\sigma^{z_1}) = P(1,1,0) + P(1,2,r_{1,2}) + P(2,1,0) + P(2,2,r_{2,2}) + \hat{P} > \lambda$,
- $p(\sigma^{z_2}) = P(1,1,r_{1,1}) + P(1,2,0) + P(2,1,r_{2,1}) + P(2,2,0) + \hat{P} > \lambda$,
- $p(\sigma^{z_3}) = P(1,1,0) + P(1,2,0) + P(2,1,r_{2,1}) + P(2,2,r_{2,2}) + \hat{P} < \lambda$,
- $p(\sigma^{z_4}) = P(1,1,r_{1,1}) + P(1,2,r_{1,2}) + P(2,1,0) + P(2,2,0) + \hat{P} < \lambda$,

which leads to the contradiction of $p(\sigma^{z_1}) + p(\sigma^{z_2}) > 2\lambda$ and $p(\sigma^{z_3}) + p(\sigma^{z_4}) < 2\lambda$, while $p(\sigma^{z_1}) + p(\sigma^{z_2}) = p(\sigma^{z_3}) + p(\sigma^{z_4})$. ∎

Let us now prove that, even if the nontrivial languages $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$ do not admit PFAs matching the state lower bound, the probabilistic model is still able to save some states with respect to nondeterminism, whose size for $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$ has been presented in Theorems 4.2.3 and 4.2.4, respectively.

Theorem 4.2.6. *The languages $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$, defined on prime powers $x_{i,j}$, for $1 \le i \le n$ and $1 \le j \le m_i$, can be recognized by PFAs with isolated cut point and $\sum_{i=1}^{n-1}\prod_{j=1}^{m_i} x_{i,j} + \sum_{j=1}^{m_n} x_{n,j}$ states.*

Proof. For the language $\mathbf{L}_{\mathrm{df}}$, we notice that every condition $\bigwedge_{j=1}^{m_i} \langle h\rangle_{x_{i,j}} = 0$, is equivalent to $\langle h\rangle_{\tau_i} = 0$, with $\tau_i = \prod_{j=1}^{m_i} x_{i,j}$. As a consequence, the language $\mathbf{L}_{\mathrm{df}}$ can be expressed as $\{\sigma^h : \bigvee_{i=1}^{n-1}\langle h\rangle_{\tau_i} = 0 \vee \bigwedge_{j=1}^{m_n}\langle h\rangle_{x_{n,j}} = 0\}$. The main idea is to associate which each product τ_i a single deterministic cycle, which checks the periodicity condition of a single one of the first $n-1$ conjunctions in the language constraint. Then we check the remaining n-th conjunction by using a set of m_n deterministic disjoint cycles of length $x_{n,j}$, for $1 \le j \le m_n$ (see Figure 4.2).

To describe formally the PFA $A_{\mathrm{df}} = \langle \varphi_{\mathrm{df}}, M_{\mathrm{df}}, \eta_{\mathrm{df}}\rangle$ for $\mathbf{L}_{\mathrm{df}}$, we first define, for every $x_{i,j}$, the vector $\eta_{i,j} = (1,0,\ldots,0) \in \mathbb{R}^{x_{i,j}}$ and, for every $\tau_i = \prod_{j=1}^{m_i} x_{i,j}$, the vector $\rho_i = (1,0,\ldots,0) \in \mathbb{R}^{\tau_i}$. The transition matrix of this automaton is $M_{\mathrm{df}} = \bigoplus_{i=1}^{n-1} C_{\tau_i} \oplus \bigoplus_{j=1}^{m_n} C_{x_{n,j}}$, where each C_x denotes the cyclic permutation of order x, the initial probability distribution is given by $\varphi_{\mathrm{df}} = \bigoplus_{i=1}^{n-1}\frac{1}{n}\rho_i \oplus \bigoplus_{j=1}^{m_n}\frac{1}{nm_n}\eta_{n,j}$, and the vector of final states is $\eta_{\mathrm{df}} = \bigoplus_{i=1}^{n-1}\rho_i \oplus \bigoplus_{j=1}^{m_n}\eta_{n,j}$. It is easy to see that A_{df} accepts the words in $\mathbf{L}_{\mathrm{df}}$ with probability at least $\frac{1}{n}$, while it accepts the words in $\mathbf{L}_{\mathrm{df}}^c$ with probability at most $\frac{1}{n} - \frac{1}{nm_n}$, so we set the cut point to $\frac{1}{n} - \frac{1}{2nm_n}$.

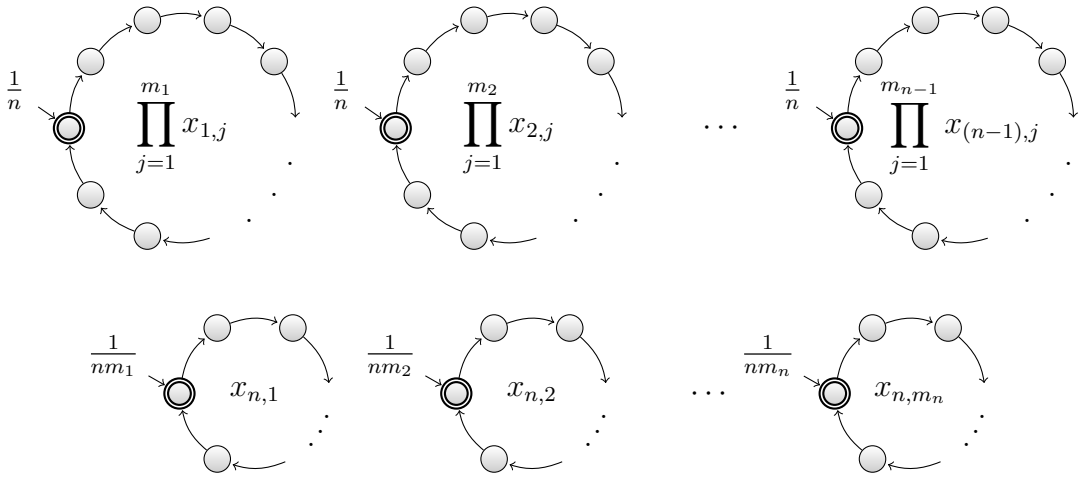


Figure 4.2: Graph representation of the PFA A_{df} for $\mathbf{L}_{\mathrm{df}}$. The number written in the center of each cycle denotes its length, i.e. the number of states in that cycle. The initial distribution is such that the cycles checking the condition $\langle h \rangle_{\tau_i} = 0$ have a higher probability than the ones checking $\langle h \rangle_{x_{n,j}} = 0$, so that the input word is accepted either if *any* of the former cycles accepts, or if *all* of the latter accept.

Now, we give the PFA $A_{\mathrm{cf}} = \langle \varphi_{\mathrm{cf}}, M_{\mathrm{cf}}, \eta_{\mathrm{cf}} \rangle$ for $\mathbf{L}_{\mathrm{cf}}$. The structure of this automaton is similar to the one of A_{df}: it is composed by $n-1$ cycles of length $\tau_i = \prod_{j=1}^{m_i} x_{i,j}$, for $1 \leq i \leq n-1$, each of those checking whether the i-th monomial is satisfied, while the last monomial is checked by m_n individual cycles of length $x_{n,j}$, for $1 \leq j \leq m_n$, as shown in Figure 4.3. The main difference with A_{df} is that the first $n-1$ cycles now have more than one accepting state, since they need to check a disjunction, in fact, in the i-th cycle, the state at distance r from the starting state is accepting iff, for some $1 \leq j \leq m_i$, it holds $\langle r \rangle_{x_{i,j}} = 0$.

More formally, the transition matrix $M_{\mathrm{cf}} = M_{\mathrm{df}}$ and the initial probability distribution $\varphi_{\mathrm{cf}} = \varphi_{\mathrm{df}}$ are as in A_{df}, while the vector of final states is $\eta_{\mathrm{cf}} = \bigoplus_{i=1}^{n-1} \bigvee_{j=1}^{m_i} \left(\bigoplus_{k=1}^{\tau_i / x_{i,j}} \eta_{i,j} \right) \oplus \bigoplus_{j=1}^{m_n} \eta_{n,j}$, where the operation $\vee$ is the componentwise $\vee$ on boolean vectors. It is not hard to see that A_{cf} accepts the words in $\mathbf{L}_{\mathrm{cf}}$ with probability at least $\frac{n-1}{n} + \frac{1}{nm_n}$, while it accepts the words in $\mathbf{L}_{\mathrm{cf}}^c$ with probability at most $\frac{n-1}{n}$, so we set the cut point to $\frac{n-1}{n} + \frac{1}{2nm_n}$.

In conclusion, in both cases we get PFAs with cut point isolated by $\frac{1}{2nm_n}$, and $\sum_{i=1}^{n-1} \tau_i + \sum_{j=1}^{m_n} x_{n,j}$ states. ∎

From the above theorems, we have that the state upper bound for PFAs is smaller than the state lower bound for NFAs for the families of languages $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$. Therefore, the probabilistic model helps reducing size with respect to

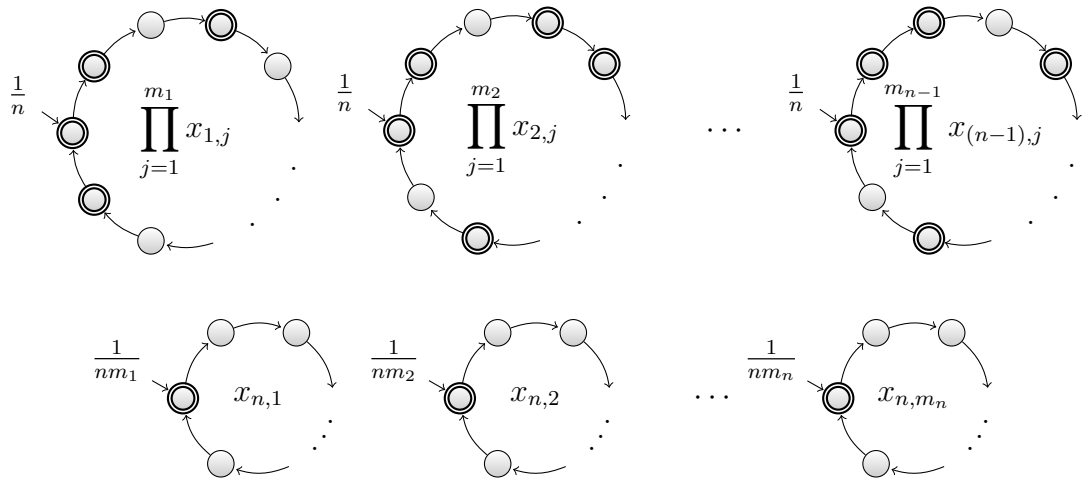


Figure 4.3: Graph representation of the PFA A_{cf} for $\mathbf{L}_{\mathrm{cf}}$. The number written in the center of each cycle denotes its length, i.e. the number of states in that cycle. As for A_{df}, the initial distribution is such that the cycles checking the periodicity condition on the first $n-1$ monomials have a higher probability than the ones individually checking the literals of the last monomial.

nondeterminism. As a natural question, one may ask for the state lower bound of PFAs for $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$. This problem turns out to be not so easy to solve. In fact, let us consider two subfamilies of these languages: let $n > 2$ and choose $n(n-1)$ powers $x_{i,j}$ of pairwise different primes, for $1 \leq i \leq n-1$ and $1 \leq j \leq n$. We define

$$\mathbf{L}_{\mathrm{df}}^{(n)} = \left\{ \sigma^h \;\middle|\; \bigvee_{i=1}^{n-1} \bigwedge_{j=1}^{n} \langle h \rangle_{x_{i,j}} = 0 \right\}, \qquad \mathbf{L}_{\mathrm{cf}}^{(n)} = \left\{ \sigma^h \;\middle|\; \bigwedge_{i=1}^{n-1} \bigvee_{j=1}^{n} \langle h \rangle_{x_{i,j}} = 0 \right\}.$$

Clearly, $\mathbf{L}_{\mathrm{df}}^{(n)}$ and $\mathbf{L}_{\mathrm{cf}}^{(n)}$ can be recognized by PFAs obtained as in the proof of Theorem 4.2.6. In the following theorem we give an alternative construction of PFAs for the recognition of these languages. Then we show that, for some choices of the prime powers $x_{i,j}$, this latter construction requires less states than the one in Theorem 4.2.6, while for other choices of $x_{i,j}$ the opposite holds.

Theorem 4.2.7. *The languages $\mathbf{L}_{\mathrm{df}}^{(n)}$ and $\mathbf{L}_{\mathrm{cf}}^{(n)}$, defined on prime powers $x_{i,j}$, for $1 \leq i \leq n-1$ and $1 \leq j \leq n$, can be recognized by PFAs with isolated cut point and $\sum_{i=1}^{n-1} \sum_{j=1}^{n} \frac{\prod_{s=1}^{n} x_{i,s}}{x_{i,j}}$ states.*

Proof. We first define the automaton for $\mathbf{L}_{\mathrm{df}}^{(n)}$ as $A = \langle \varphi, M, \eta \rangle$, where:

- $\varphi = \bigoplus_{i=1}^{n-1} \bigoplus_{j=1}^{n} \frac{1}{n(n-1)} \varphi_{i,j}$, where $\varphi_{i,j} = (1, 0, \ldots, 0) \in \mathbb{R}^{t_{i,j}}$ and $t_{i,j} = (\prod_{s=1}^{n} x_{i,s})/x_{i,j}$,

- $M = \bigoplus_{i=1}^{n-1} \bigoplus_{j=1}^{n} C_{t_{i,j}}$ where, as usual, C_x is the cyclic permutation matrix of order x,
- $\eta = \bigoplus_{i=1}^{n-1} \bigoplus_{j=1}^{n} \varphi_{i,j}$.

This automaton consists of $n-1$ sets of n disjoint deterministic cycles. The i-th conjunction of the language constraint is checked by one of these sets in the following way: the cycle of length $t_{i,j}$ checks the divisibility of the input length by all the prime powers involved in the i-th conjunction except for $x_{i,j}$. The total number of states is clearly $\sum_{i=1}^{n-1} \sum_{j=1}^{n} t_{i,j}$. We now discuss how it works. When the input word is in $\mathbf{L}_{\mathrm{df}}^{(n)}$, at least one conjunction is satisfied, i.e. there exists at least one $1 \leq i \leq n-1$ such that the input length is a multiple of all $x_{i,j}$'s, for $1 \leq j \leq n$. This implies that all the n cycles of length $t_{i,j}$, for $1 \leq j \leq n$, end their computation in an accepting state, therefore the acceptance probability is at least $\sum_{j=1}^{n} \frac{1}{n(n-1)} = \frac{1}{n-1}$. When the input word is not in $\mathbf{L}_{\mathrm{df}}^{(n)}$, none of the conjunctions is satisfied, i.e. for all $1 \leq i \leq n-1$ there exists a $1 \leq j \leq n$ such that the input length is not a multiple of $x_{i,j}$. So, all the $n-1$ cycles of length $t_{i,j}$, for $1 \leq i \leq n-1$, end their computation in a rejecting state. In this case the acceptance probability is at most $\sum_{i=1}^{n-1} \frac{1}{n(n-1)} = \frac{1}{n}$. Thus, we have that A accepts $\mathbf{L}_{\mathrm{df}}^{(n)}$ with cut point isolated by $\frac{1}{2n(n-1)}$.

For the language $\mathbf{L}_{\mathrm{cf}}^{(n)}$, we design a similar PFA, except for the final states, which are now defined by $\eta = \bigoplus_{i=1}^{n-1} \bigoplus_{j=1}^{n} \bigvee_{\substack{s=1 \\ s \neq j}}^{n} \left(\bigoplus_{k=1}^{t_{i,j}/x_{i,s}} \eta_{i,s} \right)$, where $\eta_{i,s} = (1, 0, \ldots, 0) \in \mathbb{R}^{x_{i,s}}$ and the $\vee$ operation is the componentwise $\vee$ on boolean vectors. The only difference with the previous automaton is that here each cycle of length $t_{i,j}$ checks whether the input length is a multiple of any of the prime powers involved in the i-th disjunction, except for $x_{i,j}$. Therefore, when the input word is in $\mathbf{L}_{\mathrm{cf}}^{(n)}$, in each of the $n-1$ sets of cycles at least $n-1$ cycles end their computation in an accepting state. Otherwise, there exists a set of n cycles all ending their computation in a rejecting state. Hence, the acceptance probability for a word in $\mathbf{L}_{\mathrm{cf}}^{(n)}$ is at least $\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \frac{1}{n(n-1)} = \frac{n-1}{n}$, while for a word not in $\mathbf{L}_{\mathrm{cf}}^{(n)}$ it is at most $\sum_{i=1}^{n-2} \sum_{j=1}^{n} \frac{1}{n(n-1)} = \frac{n-2}{n-1}$. Also in this case we have a cut point isolated by $\frac{1}{2n(n-1)}$. ∎

In the following examples we show that neither the construction of Theorem 4.2.6 nor the one of Theorem 4.2.7 are always optimal, in fact the choice of the prime powers is crucial to determine which of these two constructions requires less states.

Example Let L_1 be a language in the form $\mathbf{L}_{\mathrm{df}}^{(3)}$ ($\mathbf{L}_{\mathrm{cf}}^{(3)}$, resp.) where $\{x_{1,j}\}_{1 \leq j \leq 3} = \{13, 19, 23\}$ and $\{x_{2,j}\}_{1 \leq j \leq 3} = \{11, 17, 29\}$. Since disjunction (conjunction, resp.)

is commutative, we can apply the construction for PFAs of Theorem 4.2.6 in two different ways: we get as the number of states either $(13 \cdot 19 \cdot 23) + 11 + 17 + 29 = 5728$ or $(11 \cdot 17 \cdot 29) + 13 + 19 + 23 = 5478$. On the other hand, if we apply the construction of Theorem 4.2.7, we get a PFA with $19 \cdot 23 + 13 \cdot 23 + 13 \cdot 19 + 17 \cdot 29 + 11 \cdot 29 + 11 \cdot 17 = 1982$ states. In this case the second technique is clearly better.

Example Let L_2 be a language in the form either $\mathbf{L}_{\mathrm{df}}^{(3)}$ or $\mathbf{L}_{\mathrm{cf}}^{(3)}$, where $\{x_{1,j}\}_{1 \leq j \leq 3} = \{13, 19, 23\}$ and $\{x_{2,j}\}_{1 \leq j \leq 3} = \{11, 17, 3\}$. As before, we can apply the construction for PFAs of Theorem 4.2.6 in two ways, obtaining as number of states either $(13 \cdot 19 \cdot 23) + 11 + 17 + 3 = 5722$ or $(11 \cdot 17 \cdot 3) + 13 + 19 + 23 = 616$. However, if we apply the construction of Theorem 4.2.7, we get a PFA with $19 \cdot 23 + 13 \cdot 23 + 13 \cdot 19 + 17 \cdot 3 + 11 \cdot 3 + 11 \cdot 17 = 1254$ states. In this case the first technique, which allows us to have 616 states, is better.

4.2.1 Particular cases

Here, we consider special families of $\mathbf{L}_{\mathrm{df}}$ and $\mathbf{L}_{\mathrm{cf}}$ languages, and compare the descriptional complexity of their PFAs and NFAs. For these families we can give a lower bound also on the size of the minimal PFAs.

Let $x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m} > 1$ be pairwise different prime powers, for some $n, m > 0$. We define the languages:

$$\mathbf{L}_{\vee\wedge} = \left\{ \sigma^h \mid \bigvee_{i=1}^{n} \langle h \rangle_{x_i} = 0 \vee \left(\bigwedge_{i=n+1}^{n+m} \langle h \rangle_{x_i} = 0 \right) \right\},$$
$$\mathbf{L}_{\wedge\vee} = \left\{ \sigma^h \mid \bigwedge_{i=1}^{n} \langle h \rangle_{x_i} = 0 \wedge \left(\bigvee_{i=n+1}^{n+m} \langle h \rangle_{x_i} = 0 \right) \right\}.$$

Notice that the language $\mathbf{L}_{\vee\wedge}$ ($\mathbf{L}_{\wedge\vee}$, resp.) is in the form $\mathbf{L}_{\mathrm{df}}$ ($\mathbf{L}_{\mathrm{cf}}$, resp.), with the first n conjunctions (disjunctions, resp.) consisting of a single periodicity condition. It is easy to verify that both $\mathbf{L}_{\vee\wedge}$ and $\mathbf{L}_{\wedge\vee}$ are properly $\prod_{i=1}^{n+m} x_i$-cyclic.

For these families of languages, nondeterminism can help in saving states with respect to determinism, but not as much as probabilism. Indeed, from Theorem 4.2.3, we can directly obtain that the languages $\mathbf{L}_{\vee\wedge}$ and $\mathbf{L}_{\wedge\vee}$ admit minimal NFAs having $\sum_{i=1}^{n} x_i + \prod_{j=n+1}^{n+m} x_j$ and $\prod_{i=1}^{n} x_i \cdot \sum_{j=n+1}^{n+m} x_j$ states, respectively. On the other hand, we know from [41] that PFAs accepting $\mathbf{L}_{\vee\wedge}$ and $\mathbf{L}_{\wedge\vee}$ with isolated cut point cannot have less than $\sum_{i=1}^{n+m} x_i$ states. This lower bound, together with Theorem 4.2.6, leads to the following

Theorem 4.2.8. *The languages* $\mathbf{L}_{\vee\wedge}$ *and* $\mathbf{L}_{\wedge\vee}$ *defined on prime powers* x_i*, for* $1 \leq i \leq n+m$*, admit minimal* PFA*s with isolated cut point and* $\sum_{i=1}^{n+m} x_i$ *states.*

It may be interesting to observe that we have exhibited a family of languages for which the constructions of Theorem 4.2.6 become optimal, since they match the lower bound. Notice that the same lower bound cannot be reached by the constructions of PFAs in Theorem 4.2.3, except for a simple disjunction or conjunction of two periodicity conditions.

Chapter 5

Quantum automata

In this chapter we present another type of finite state machine, called *quantum automaton* (QFA for short), which represents the mathematical abstraction of a computing device whose computation is the result of a quantum physical phenomenon.

Several variants of quantum automata have been proposed in the literature, many of them differ in the measurement policy: in what follows we will present the *Measure-Once* model, where the observation on the system is performed only once, when the whole computation is finished, and the *Measure-Many* model, where the system is observed at every step of computation. We also consider a third variant of QFAs with *control language*, which are a hybrid model, made of a quantum computational core with a classical response on the result of measurements.

For each of these models, we formally define the behavior in terms of *induced event*, and *language recognized* with isolated cut point. We discuss the computational power of the presented variants of automata and properties of the families of languages accepted by such automata. Finally, we give an explicit construction of MM-QFAs for recognizing any unary regular language, which requires a state complexity linear in the size of the original automaton. This construction was published in [16, 17]. We remark that having a linear size conversion from DFA to MM-QFA is not trivial, since it was shown in [1] the existence of a family of languages L_n, recognized by an n-state DFA, while any MM-QFA recognizing L_n with some constant probability greater than $\frac{1}{2}$ has $2^{\Omega(n/\log n)}$ states.

5.1 Measure-Once quantum automata

A *one-way Measure-Once quantum finite state automaton* (MO-QFA) is a tuple

$$\Omega \;=\; \langle\, \Sigma,\;\; Q,\;\; \Psi_0,\;\; \tau,\;\; F \,\rangle$$

where

- Σ is the (finite) input alphabet,
- Q is the (finite) set of *basis states*,
- Ψ_0 is the *starting state*,
- $F \subseteq Q$ is the set of accepting states,
- τ is the transition function such that

$$\tau : Q \times \Sigma \times Q \to \mathbb{C}.$$

The value $\tau(q_j, \sigma, q_k)$ represents the *complex amplitude* associated to the transition from state q_j to state q_j upon reading the symbol σ. The amplitudes determine the stochastic evolution of the system in the following sense: if we observe the system in state q_j and the next input symbol is σ, the probability of observing the system in state q_k at the following step is $|\tau(q_j, \sigma, q_k)|^2$. Since amplitudes define a probability distribution, the following property must hold

$$\sum_{q \in Q} |\tau(p, \sigma, q)|^2 = 1,$$

for each $p \in Q$ and $\sigma \in \Sigma$. In this model though, the only observation is performed at the end of the computation: if the observed state is in F, then the automaton *accepts* the input word, otherwise it *rejects* it.

Since the transitions are nondeterministic, during its computation a QFA may be in more than one state at the same time, therefore the configuration of the system is represented by a unitary vector in the m-dimensional Hilbert space, where $m = |Q|$, called *quantum superposition*:

$$\varphi = \sum_{j=1}^{m} \alpha_j e_j$$

where, as usual, e_j is the characteristic vector of the basis state q_j. The vectors e_j denote the canonical base for $\mathbb{C}^m$, while the complex coefficients α_j are the amplitudes of the corresponding states, for which it holds

$$\sum_{j=1}^{m} \alpha_j^2 = 1$$

which ensures that the vector φ is unitary.

The transition function can be expressed through a set of unitary matrices $\{U(\sigma)\}_{\sigma\in\Sigma}$ such that

$$(U(\sigma))_{j,k} = \tau(q_j, \sigma, q_k).$$

Upon reading the input symbol σ, the automaton Ω changes its configuration from φ to

$$\varphi' = \varphi U(\sigma).$$

The observation, which determines whether a word is accepted or not, is represented by two matrices $P(a)$ and $P(r)$ such that

$$(P(a))_{j,k} = \begin{cases} 1 & \text{if } j = k \text{ and } j \in F, \\ 0 & \text{otherwise,} \end{cases} \quad (P(r))_{j,k} = \begin{cases} 1 & \text{if } j = k \text{ and } j \notin F, \\ 0 & \text{otherwise,} \end{cases}$$

which project the current superposition φ on the two orthogonal subspaces of $\mathbb{C}^m$ spanned, respectively, by the characteristic vectors of the basic accepting and rejecting states.

This leads to the matrix representation of the MO-QFA Ω:

$$\Omega = \langle \varphi_0, \{U(\sigma)\}_{\sigma\in\Sigma}, \mathcal{O} \rangle,$$

where $\mathcal{O}$ is a Hermitian matrix characterized by the orthogonal projectors $P(a)$, $P(r)$ and the associated eigenvalues λ_a, λ_r, so that it holds

$$\mathcal{O} = \lambda_a P(a) + \lambda_r P(r),$$

with $P(a) + P(r) = I$. An observation on the superposition φ has result λ_x, with $x \in \{a, r\}$ with probability $\|\varphi P(x)\|^2$. Since the matrix $P(r)$ can always be univocally obtained from $P(a)$, we will substitute $\mathcal{O}$ with the accepting projector $P(a)$ in the matrix representation of MO-QFAs.

The behavior of the automaton Ω on the input word $\omega = \sigma_1\sigma_2\cdots\sigma_n$ is described by the probability of observing Ω in an accepting state at the end of the computation on ω:

$$\mathcal{P}_\Omega(\omega) = \|\varphi_0 \prod_{j=1}^{n} U(\sigma_j) P(a)\|^2.$$

The function $\mathcal{P}_\Omega$ is called the *event induced by the* MO-QFA Ω.

The language recognized by Ω *with cut point* $\lambda \in [0, 1]$ is defined as

$$L_\Omega = \{\omega \in \Sigma^* \mid \mathcal{P}_\Omega(\omega) > \lambda\},$$

and we say that λ is *isolated by* δ if

$$\mathcal{P}_\Omega(\omega) \begin{cases} \geq \lambda + \delta & \text{if } \omega \in L_\Omega \\ \leq \lambda - \delta & \text{if } \omega \notin L_\Omega. \end{cases}$$

5.2 Measure-Many quantum automata

We now present a second model of QFA, where measurements are performed not only at the end of the computation, but also at every single step. In this case, we augment the input with a right end-marker #. This model of QFA is called *one-way Measure-Many finite state automaton* (MM-QFA) and, as for MO-QFAs, we are going to define it through its matrix representation, given by the tuple

$$\Omega = \langle \varphi_0, \{U(\sigma)\}_{\sigma \in \Sigma \cup \{\#\}}, \mathcal{O} \rangle,$$

where $\mathcal{O} = \lambda_a P(a) + \lambda_r P(r) + \lambda_g P(g)$, for orthogonal projectors $P(a), P(r), P(g)$ such that $P(a) + P(r) + P(g) = I$, is the observable applied at *every step* of the computation. If the result of the observation is the eigenvalue λ_a, the automaton halts its computation and accepts the input word, while if λ_r is observed, Ω halts and rejects the input word. Finally, if λ_g is observed, then the computation continues. After performing a measurement on the superposition φ, the quantum configuration of the system can be irreversibly modified: the new configuration is expressed by the normalized projection of φ on the eigenspace corresponding to the observed eigenvalue

$$\varphi' = \frac{1}{\|\varphi P(g)\|} \varphi P(g). \tag{5.1}$$

The normalization is performed only when the automaton does not end its computation. Given the input string $\omega = \sigma_1 \sigma_2 \cdots \sigma_n$, if at time $m \leq n$ Ω has not yet performed a halting projection (either accepting or rejecting), then its current superposition is

$$\varphi_m = \frac{\varphi_0 \prod_{j=1}^{m} (U(\sigma_j) P(g))}{\left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j) P(g)) \right\|}. \tag{5.2}$$

We can prove the above property by induction: Equation (5.1) implies that the configuration of Ω at the first step of computation is

$$\varphi_1 = \frac{\varphi_0 (U(\sigma_1) P(g)}{\|\varphi_0 U(\sigma_1) P(g)\|}$$

while, assuming Equation (5.2) holds for φ_{m-1}, we have

$$\begin{aligned}
\varphi_m =& \frac{\varphi_{m-1} U(\sigma_m) P(g)}{\|\varphi_{m-1} U(\sigma_m) P(g)\|} \\
=& \frac{\varphi_0 \prod_{j=1}^{m-1} (U(\sigma_j) P(g)) \cdot U(\sigma_m) P(g)}{\left\| \varphi_0 \prod_{j=1}^{m-1} (U(\sigma_j) P(g)) \right\|} \cdot \left\| \frac{\left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j) P(g)) \right\|}{\varphi_0 \prod_{j=1}^{m} (U(\sigma_j) P(g)) \cdot U(\sigma_m) P(g)} \right\| \\
=& \frac{\varphi_0 \prod_{j=1}^{m} (U(\sigma_j) P(g))}{\left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j) P(g)) \right\|}.
\end{aligned}$$

The probability that Ω, at time m, has not yet ended its computation is

$$\left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j)P(g)) \right\|^2,$$

therefore Ω accepts ω at computational step $m+1$ with probability

$$\begin{aligned}
&\left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j)P(g)) \right\|^2 \cdot \|\varphi_m U(\sigma_{m+1})P(g)\|^2 = \\
&= \left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j)P(g)) \right\|^2 \cdot \left\| \frac{\varphi_0 \prod_{j=1}^{m} (U(\sigma_j)P(g)) \cdot U(\sigma_{m+1})P(g)}{\left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j)P(g)) \right\|} \right\|^2 = \\
&= \left\| \varphi_0 \prod_{j=1}^{m} (U(\sigma_j)P(g)) \cdot U(\sigma_{m+1})P(g) \right\|^2.
\end{aligned}$$

The above considerations imply that the probability of Ω accepting an input string $\omega = \sigma_1\sigma_2\cdots\sigma_n$ is

$$\begin{aligned}
\mathcal{P}_\Omega(\omega) = &\sum_{k=1}^{n} \left\| \varphi_0 \left(\prod_{j=1}^{k-1} (U(\sigma_j)P(g)) \right) U(\sigma_k)P(a) \right\|^2 + \\
&+ \left\| \varphi_0 \left(\prod_{j=1}^{n} (U(\sigma_j)P(g)) \right) U(\#)P(a) \right\|^2.
\end{aligned}$$

As before, the language recognized by the MM-QFA Ω *with cut point* $\lambda \in [0,1]$ is

$$L_\Omega = \{\omega \in \Sigma^* \mid \mathcal{P}_\Omega(\omega) > \lambda\},$$

and λ is *isolated by* δ if

$$\mathcal{P}_\Omega(\omega) \begin{cases} \geq \lambda + \delta & \text{if } \omega \in L_\Omega \\ \leq \lambda - \delta & \text{if } \omega \notin L_\Omega. \end{cases}$$

Without loss of generality, we can assume that the MM-QFAs we consider have two distinct observables, depending on whether the last read symbol is in Σ or is the end-marker #, i.e. in the form

$$\Omega = \langle \varphi_0, \{U(\sigma)\}_{\sigma \in \Sigma \cup \{\#\}}, \mathcal{O}_{int}, \mathcal{O}_{fin} \rangle, \tag{5.3}$$

where $\mathcal{O}_{int} = \lambda_a P_{int}(a) + \lambda_r P_{int}(r) + \lambda_g P_{int}(g)$ is the observable on the system after the evolution on a symbol $\sigma \in \Sigma$, while $\mathcal{O}_{fin} = \lambda_a P_{fin}(a) + \lambda_r P_{fin}(r)$ describes

the observation after the evolution $U(\#)$. The acceptance probability of a word $\omega = \sigma_1\sigma_2\cdots\sigma_n$ is

$$\mathcal{P}_\Omega(\omega) = \sum_{k=1}^{n} \left\| \varphi_0 \left(\prod_{j=1}^{k-1} U(\sigma_j) P_{int}(g) \right) U(\sigma_k) P_{int}(a) \right\|^2 + \\ + \left\| \varphi_0 \left(\prod_{j=1}^{n} U(\sigma_j) P_{int}(g) \right) U(\#) P_{fin}(a) \right\|^2 .$$

In fact, the following result holds

Theorem 5.2.1. *Every* MM-QFA *with distinct intermediate and final observables and m states can be simulated by a* MM-QFA *with a single observable and $2m$ states.*

Proof. Assume Ω is a MM-QFA with m states defined as in Equation (5.3). We construct an equivalent MM-QFA Ω' defined as

$$\Omega' = \langle \varphi_0', \{U'(\sigma)\}_{\sigma\in\Sigma\cup\{\#\}}, \mathcal{O}' \rangle,$$

such that

- $\varphi_0' = \varphi_0 \oplus \zeta \in \mathbb{C}^{2m}$, where $\zeta = (0, 0, \ldots, 0) \in \mathbb{C}^m$,
- $U'(\sigma) = \begin{cases} \begin{pmatrix} U(\sigma) & 0 \\ 0 & I \end{pmatrix} & \text{if } \sigma \in \Sigma \\ \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \cdot \begin{pmatrix} I & 0 \\ 0 & U(\#) \end{pmatrix} & \text{if } \sigma = \#, \end{cases}$
  with $U'(\sigma) \in \mathbb{C}^{2m\times 2m}$,
- $\mathcal{O}' = \lambda_a P'(a) + \lambda_r P'(r) + \lambda_g P'(g)$ is such that
  $P'(a) = P_{int}(a) \oplus P_{fin}(a)$,
  $P'(r) = P_{int}(r) \oplus P_{fin}(r)$,
  $P'(g) = P_{int}(g) \oplus Z$, where Z is the $m \times m$ null matrix.

One can easily verify that, as long as symbols in Σ are read, the whole computation of Ω' is restricted to the first m components, where it works exactly like Ω. When the end-marker # is read, the matrix $U'(\#)$ moves the computation from the first m components to the last m components, where the projectors of $\mathcal{O}_{fin}$ operate. ∎

It is not hard to see that, by restricting $\mathcal{O}_{int}$ to be described only by $P_{int}(g)$, i.e. if $P_{int}(g) = I$, the resulting MM-QFA is actually a MO-QFA. Another particular

subclass of MM-QFAs is the *end decisive* MM-QFAs [21], where the intermediate observable $\mathcal{O}_{int}$ is composed only by projectors $P_{int}(g)$ and $P_{int}(r)$, i.e. every word can only be accepted after the end-marker # is read.

5.3 Quantum automata with control languages

The last variant of QFA we consider has been proposed in [10] as a hybrid device: in this model, an observable $\mathcal{O}$ with a fixed, but arbitrary, set of possible results $C = \{c_1, c_2, \ldots, c_s\}$ is considered. On any given input word $x \in \Sigma^*$, the computation displays a sequence $y \in C^*$ of resulting measurements of $\mathcal{O}$ with a certain probability $p(y; x)$. The computation is accepting if and only if y belongs to a fixed regular control language $\mathcal{L} \subseteq C^*$.

More formally, given an alphabet Σ and an end-marker symbol $\# \notin \Sigma$, an (m, k)-*state one-way quantum finite automaton with control language* (QFC) is a system

$$\Omega = \langle \varphi_0, \{U(\sigma)\}_{\sigma \in \Gamma}, \mathcal{O}, \mathcal{L} \rangle,$$

where

- $\Gamma = \Sigma \cup \{\#\}$,
- $\varphi_0 \in \mathbb{C}^m$ is the initial superposition of the quantum finite control, satisfying $\|\varphi_0\| = 1$,
- $U(\sigma) \in \mathbb{C}^{m \times m}$, for all $\sigma \in \Gamma$,
- $\mathcal{O}$ is an observable on $\mathbb{C}^m$: if $C = \{c_1, c_2, \ldots, c_s\}$ is the class of all possible results of measurements of $\mathcal{O}$, then $P(c_j)$ denotes the projector on the eigenspace corresponding to c_j, for all $c_j \in C$,
- $\mathcal{L} \subseteq C^*$ is the regular *control language*, recognized by a k-state DFA.

We now define the behavior of Ω on a word $x_1 x_2 \cdots x_n \in \Gamma^*$. The quantum finite control works like in a MM-QFA: the computation starts in the state φ_0, then the evolutions associated with the symbols $x_1, x_2, \ldots, x_n$ are applied in succession. The evolution corresponding to a symbol $\sigma \in \Gamma$ consists of two steps:

1. First, $U(\sigma)$ is applied to the current state φ of the automaton, yielding the new state φ'.
2. Then, the observable $\mathcal{O}$ is measured on φ': the result of this measurement is c_j with probability $\|\varphi P(c_j)\|$, and the state of the automaton collapses to $\varphi' P(c_j)/\|\varphi' P(c_j)\|$.

Thus, a computation on $x_1x_2\cdots x_n$ leads to a sequence $y_1y_2\cdots y_n$ of results of the observation on $\mathcal{O}$ with probability

$$p(y_1y_2\cdots y_n; x_1x_2\cdots x_n) = \left\| \varphi_0 \prod_{j=1}^{n} (U(x_j)P(y_j)) \right\|^2 .$$

A computation leading to the word $y_1y_2\cdots y_n$ is *accepting* if $y_1y_2\cdots y_n \in \mathcal{L}$, otherwise it is rejecting. Hence, the probability that, on input $x_1x_2\cdots x_n$, the automaton Ω generates an accepting computation is

$$\psi_\Omega(x_1x_2\cdots x_n) = \sum_{y_1y_2\cdots y_n \in \mathcal{L}} p(y_1y_2\cdots y_n; x_1x_2\cdots x_n).$$

In what follows, we are interested in the behavior of Ω on words in $\Sigma^*\#$, therefore, the stochastic event $\mathcal{P}_\Omega : \Sigma^* \to [0,1]$ induced by Ω is

$$\mathcal{P}_\Omega(x_1x_2\cdots x_n) = \psi_\Omega(x_1x_2\cdots x_n\#),$$

i.e., the probability that $x_1x_2\cdots x_n\#$ generates an accepting computation.

As usual, the language recognized by the QFC Ω *with cut point* $\lambda \in [0,1]$ is

$$L_\Omega = \{\omega \in \Sigma^* \mid \mathcal{P}_\Omega(\omega) > \lambda\},$$

and λ is *isolated by* δ if

$$\mathcal{P}_\Omega(\omega) \begin{cases} \geq \lambda + \delta & \text{if } \omega \in L_\Omega \\ \leq \lambda - \delta & \text{if } \omega \notin L_\Omega. \end{cases}$$

5.4 Computational power and properties of quantum automata

In this Section we recall the main known results on the computational power of the quantum devices we have presented in this Chapter. We denote with L_{MO}, L_{MMe}, L_{MM} and L_{qfc} the classes of languages recognized with isolated cut point by MO-QFAs, end-decisive MM-QFAs, MM-QFAs and QFCs, respectively. The computational power of the Measure-Once model on a generic alphabet has been fully characterized in [7] by Bertoni and Carpentieri as follows

Theorem 5.4.1. *L_{MO} coincides with the class of reversible regular languages.*

Unlike L_{MO}, the class of languages recognized by MM-QFAs with isolated cut point is still unknown. However, it was proved in [37] that MM-QFAs can only recognize regular language, but cannot recognize, for example, the language denoted by the regular expression $\{a,b\}^*a$. This leads to the following

Theorem 5.4.2. *The class L_{MM} is a proper subclass of regular languages.*

Although it is hard to characterize the class of languages recognized by MM-QFAs, there are some *forbidden construction* which guarantee that a regular language does not belong to the class L_{MM}. In particular, the following property is shown in [21]:

Theorem 5.4.3. *Let $L \in \Sigma^*$ be a regular language and A the minimal* DFA *recognizing L and containing the pattern in figure 5.1, where p, q are states of A and $v, w \in \Sigma^*$. Then, $L \notin L_{MM}$.*

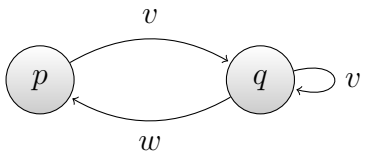


Figure 5.1: Forbidden pattern for MM-QFAs.

Another example of critical pattern is shown in [5]:

Theorem 5.4.4. *Let $L \in \Sigma^*$ be a regular language and A the minimal* DFA *recognizing L and containing the pattern in figure 5.2, where p, q, s are states of A and $v, w, t, t' \in \Sigma^*$. Then, $L \notin L_{MM}$.*

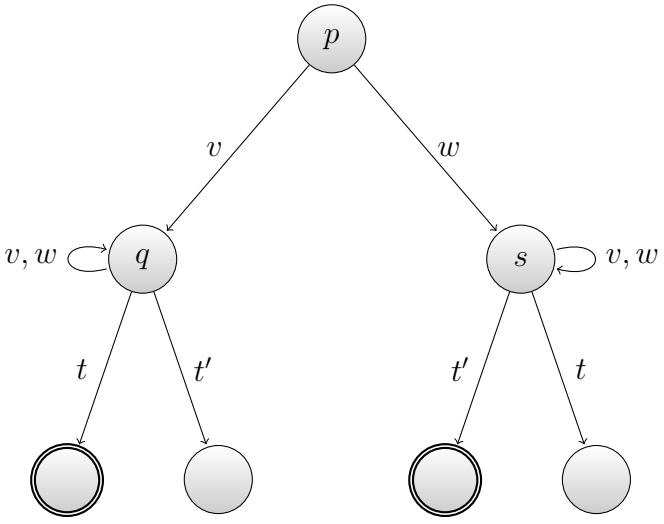


Figure 5.2: Forbidden pattern for MM-QFAs.

In [5], the authors present two languages L_1 and L_2, which can be recognized by MM-QFAs with isolated cut point, such that the minimal DFA for $L_1 \cup L_2$ requires the pattern in Figure 5.1, thus showing that L_{MM} is not closed under

union. This implies that the constraint of end-decisiveness strictly reduces the class of recognized languages, since the class L_{MMe} is closed under union [21].

Among the three presented variants of QFA, the most powerful is the model of QFA with control language, whose class of recognized languages was shown in [39] to be the class of regular languages:

Theorem 5.4.5. *[39] L_{qfc} coincides with the class of regular language.*

Moreover, the following Theorem shows that any MM-QFA can always be simulated by a QFC without increasing the number of quantum states and using only 3 classical states:

Theorem 5.4.6. *[10] For every m-state* MM-QFA Ω*, there exists a $(m, 3)$-state* QFC Ω' *with possible results of measurements $C = \{a, r, g\}$ and control language $g^*a\{a, r, g\}^*$, such that $\mathcal{P}_\Omega = \mathcal{P}_{\Omega'}$.*

The above considerations can be summarized by the diagram in Figure 5.3, showing the relations among different classes of languages.

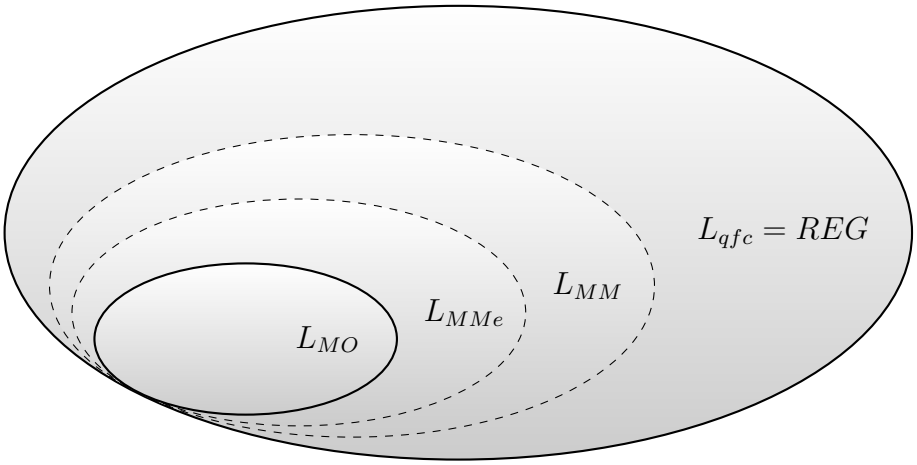


Figure 5.3: Diagram showing the relations among the computational power of the different variants of quantum automata. *REG* denotes the class of regular languages, while the dashed line indicates that the class of languages is still not fully characterized.

For an event χ, we call $\bar{\chi} = 1 - \chi$ the *complementary event* of χ. In what follows, we show that the events induced by MO-QFAs are closed under complement, convex combination and product:

Theorem 5.4.7. *Given a* MO-QFA $\langle \varphi_0, U(\sigma), P \rangle$*, one can always find a* MO-QFA $\bar{\Omega}$ *such that, for any word $\omega \in \Sigma^*$,*

$$\mathcal{P}_{\bar{\Omega}}(\omega) = 1 - \mathcal{P}_\Omega(\omega).$$

In other words, $\bar{\Omega}$ induces the event complementary to the one induced by Ω.

Proof. To obtain the MO-QFA $\bar{\Omega}$ it is sufficient to replace the accepting projector P with its orthogonal $\bar{P} = I - P$:

$$\bar{\Omega} = \langle\ \varphi_0,\ U(\sigma),\ \bar{P} = I - P\ \rangle.$$

∎

Theorem 5.4.8. *Given a sequence of* MO-QFA*s* $\Omega_1, \Omega_2, \ldots, \Omega_m$ *defined as*

$$\Omega_j = \langle\ \varphi_j,\ \{U_j(\sigma)\}_{\sigma\in\Sigma},\ P_j\ \rangle,$$

for any $\alpha_1, \alpha_2, \ldots, \alpha_m \in [0,1]$ *such that* $\sum_{j=1}^{m} \alpha_j = 1$*, it is always possible to construct, through the direct sum operation, a* MO-QFA

$$\Omega_{\oplus} = \langle \bigoplus_{j=1}^{m} \sqrt{\alpha_j}\varphi_j,\ \left\{ U_{\oplus}(\sigma) = \bigoplus_{j=1}^{m} U_j(\sigma) \right\}_{\sigma\in\Sigma},\ \bigoplus_{j=1}^{m} P_j\ \rangle$$

inducing the event determined by the weighted average of the events induced by the MO-QFA*s* Ω_j*, with* α_j *as coefficient. Therefore, for each word* $\omega \in \Sigma^*$ *it holds*

$$\mathcal{P}_{\Omega_{\oplus}}(\omega) = \sum_{j=1}^{m} \alpha_j \mathcal{P}_{\Omega_j}(\omega).$$

Proof. We prove the result in the unary case $\Sigma = \{\sigma\}$, the generalization to an alphabet of arbitrary size is trivial. Let $A = \langle \varphi_A, U_A, P_A \rangle$ e $B = \langle \varphi_B, U_B, P_B \rangle$ be two MO-QFAs with associated events

$$\mathcal{P}_A(\sigma^k) = \left\| \varphi_A \cdot {U_A}^k \cdot P_A \right\|^2,$$
$$\mathcal{P}_B(\sigma^k) = \left\| \varphi_B \cdot {U_B}^k \cdot P_B \right\|^2.$$

For $\alpha \in [0,1]$, the MO-QFA

$$C_{\oplus} \ = \ \langle \sqrt{\alpha}\varphi_A \oplus \sqrt{(1-\alpha)}\varphi_B, U_A \oplus U_B, P_A \oplus P_B \rangle$$

induces the event

$$\mathcal{P}_{C_{\oplus}}(\sigma^k) = \alpha \mathcal{P}_A(\sigma^k) + (1-\alpha)\mathcal{P}_B(\sigma^k),$$

indeed

$$\begin{aligned}
\mathcal{P}_{C_{\oplus}}(\sigma^k) =& \left\| (\sqrt{\alpha}\varphi_A \oplus \sqrt{(1-\alpha)}\varphi_B)(U_A \oplus U_B)^k\,(P_A \oplus P_B) \right\|^2 \\
=& \left\| \left(\sqrt{\alpha}\varphi_A \oplus \sqrt{(1-\alpha)}\varphi_B \right) \left({U_A}^k \oplus {U_B}^k \right) (P_A \oplus P_B) \right\|^2 \\
=& \left\| \sqrt{\alpha}(\varphi_A {U_A}^k P_A) \oplus \sqrt{(1-\alpha)}(\varphi_B {U_B}^k P_B) \right\|^2 \\
=& \left\| \sqrt{\alpha}\varphi_A {U_A}^k P_A \right\|^2 + \left\| \sqrt{(1-\alpha)}\varphi_B {U_B}^k P_B \right\|^2 \\
=& \alpha \mathcal{P}_A(\sigma^k) + (1-\alpha)\mathcal{P}_B(\sigma^k).
\end{aligned}$$

∎

Theorem 5.4.9. *Given a sequence of* MO-QFA*s* $\Omega_1, \Omega_2, \ldots, \Omega_m$ *defined as*

$$\Omega_j = \langle\ \varphi_j,\ \{U_j(\sigma)\}_{\sigma\in\Sigma},\ P_j\ \rangle,$$

it is always possible to construct, through the Kronecker product, a MO-QFA

$$\Omega_\otimes = \langle\ \bigotimes_{j=1}^{m}\varphi_j,\ \left\{U_\otimes(\sigma) = \bigotimes_{j=1}^{m} U_j(\sigma)\right\}_{\sigma\in\Sigma},\ \bigotimes_{j=1}^{m} P_j\ \rangle$$

inducing the product of the events induced by the MO-QFA*s* Ω_j*. Therefore, for every word* $\omega \in \Sigma^*$*, it holds*

$$\mathcal{P}_{\Omega_\otimes}(\omega) = \prod_{j=1}^{m} \mathcal{P}_{\Omega_j}(\omega).$$

Proof. Also in this case, we prove the result for the unary alphabet $\Sigma = \{\sigma\}$. Consider the MO-QFAs A and B defined in the proof of Theorem 5.4.8. We show that the MO-QFA

$$C_\otimes \ = \ \langle (\varphi_A \otimes \varphi_B), U_A \otimes U_B, P_A \otimes P_B\rangle$$

induces the event

$$\mathcal{P}_{C_\otimes}(\sigma^k) = \mathcal{P}_A(\sigma^k)\cdot\mathcal{P}_B(\sigma^k),$$

indeed:

$$\begin{aligned}
\mathcal{P}_{C_\otimes}(\sigma^k) =& \left\| (\varphi_A\otimes\varphi_B)\,(U_A\otimes U_B)^k\,(P_A\otimes P_B)\right\|^2\\
=& \left\| (\varphi_A\otimes\varphi_B)\left({U_A}^k\otimes {U_B}^k\right)(P_A\otimes P_B)\right\|^2\\
=& \left\| \left(\varphi_A {U_A}^k P_A\ \otimes\ \varphi_B {U_B}^k P_B\right)\right\|^2\\
=& \left\|\varphi_A {U_A}^k P_A\right\|^2 \cdot \left\|\varphi_B {U_B}^k P_B\right\|^2\\
=& \mathcal{P}_A(\sigma^k)\cdot\mathcal{P}_B(\sigma^k).
\end{aligned}$$

∎

We now discuss the closure properties of the Measure-Many model, presenting the generalization of Theorems 5.4.7 and 5.4.8 to MM-QFAs.

Theorem 5.4.10. *For any* MM-QFA $\Omega = \langle\ \varphi,\ \{U(\sigma)\}_{\sigma\in(\Sigma\cup\{\#\})},\ \mathcal{O}_{int}\ \mathcal{O}_{fin}\ \rangle$*, there exists a* MM-QFA $\bar{\Omega}$ *such that*

$$\mathcal{P}_{\bar{\Omega}}(\omega) = 1 - \mathcal{P}_\Omega(\omega)$$

for every word $\omega \in \Sigma^*$*, i.e.,* $\bar{\Omega}$ *induces the event complementary to the one of* Ω*.*

Proof. In order to obtain the MM-QFA $\bar{\Omega}$, it is sufficient to swap the accepting and rejecting projectors: let $O_{int} = \lambda_a P_{int}(a) + \lambda_r P_{int}(r) + \lambda_g P_{int}(g)$ and $O_{fin} = \lambda_a P_{fin}(a) + \lambda_r P_{fin}(r)$ be the two observables of Ω, then the automaton

$$\bar{\Omega} = \langle\ \varphi,\ \{U(\sigma)\}_{\sigma\in(\Sigma\cup\{\#\})}\ ,\ \bar{\mathcal{O}}_{int}\ \bar{\mathcal{O}}_{fin}\ \rangle,$$

where $\bar{\mathcal{O}}_{int} = \lambda_a P_{int}(r) + \lambda_r P_{int}(a) + \lambda_g P_{int}(g)$ and $\bar{\mathcal{O}}_{fin} = \lambda_a P_{fin}(r) + \lambda_r P_{fin}(a)$, induces the event complementary to the one of Ω. ∎

Theorem 5.4.11. *Given a sequence of* MM-QFA*s* $\Omega_1, \Omega_2, \ldots, \Omega_m$ *defined as*

$$\Omega_j = \langle\ \varphi_j,\ \{U_j(\sigma)\}_{\sigma\in(\Sigma\cup\{\#\})}\ ,\ \mathcal{O}_j\ \rangle,$$

the automaton

$$\Omega_{\oplus} = \langle\ \bigoplus_{j=1}^{m}\sqrt{\alpha_j}\varphi_j,\ \left\{\bigoplus_{j=1}^{m}U_j(\sigma)\right\}_{\sigma\in(\Sigma\cup\{\#\})}\ ,\ \bigoplus_{j=1}^{m}\mathcal{O}_j\ \rangle,$$

where $\sum_{j=1}^{m}\alpha_j = 1$*, induces the convex combination of the events induced by* $\Omega_1, \Omega_2, \ldots, \Omega_m$*, with* $\alpha_1, \alpha_2, \ldots, \alpha_m$ *as coefficients. More formally, for any word* $\omega \in \Sigma^*$*, it holds*

$$\mathcal{P}_{\Omega_{\oplus}}(\omega) = \sum_{j=1}^{m}\alpha_j\mathcal{P}_{\Omega_j}(\omega).$$

Proof. Consider the two MM-QFAs $A = \langle\varphi_A,\ \{U_A(\sigma)\}_{\sigma\in(\Sigma\cup\{\#\})}\ ,\ \mathcal{O}_A\rangle$ and $B = \langle\varphi_B,\ \{U_B(\sigma)\}_{\sigma\in(\Sigma\cup\{\#\})}\ ,\ \mathcal{O}_B\rangle$, whose events induced on input $\omega = a_1\cdots a_n$ are, respectively,

$$\begin{aligned}\mathcal{P}_A(\omega) = &\sum_{k=1}^{n}\left\|\varphi_A\left(\prod_{j=1}^{k-1}U_A(a_j)P_A(g)\right)U(a_k)P_A(a)\right\|^2 + \\ &+\left\|\varphi_0\left(\prod_{j=1}^{n}U(a_j)P_A(g)\right)U(\#)P_A(a)\right\|^2,\\ \mathcal{P}_B(\omega) = &\sum_{k=1}^{n}\left\|\varphi_B\left(\prod_{j=1}^{k-1}U_B(a_j)P_B(g)\right)U(a_k)P_B(a)\right\|^2 + \\ &+\left\|\varphi_0\left(\prod_{j=1}^{n}U(a_j)P_B(g)\right)U(\#)P_B(a)\right\|^2.\end{aligned}$$

The automaton

$$C_{\oplus}\ =\ \langle\left(\sqrt{\alpha}\cdot\varphi_A\oplus\sqrt{1-\alpha}\cdot\varphi_B\right),\{U_A(\sigma)\oplus U_B(\sigma)\}_{\sigma\in(\Sigma\cup\{\#\})},\mathcal{O}_A\oplus\mathcal{O}_B\rangle$$

induces the event

$$\mathcal{P}_{C_\oplus}(\omega) = \alpha \cdot \mathcal{P}_A(\omega) + (1-\alpha) \cdot \mathcal{P}_B(\omega),$$

representing the convex linear combination of the events induced by A and B, indeed:

$$\begin{aligned}\mathcal{P}_{C_\oplus}(\omega) = &\sum_{k=1}^{n} \| \left(\sqrt{\alpha} \cdot \varphi_A \oplus \sqrt{1-\alpha} \cdot \varphi_B\right) \cdot \prod_{j=1}^{k-1} \left[\left(U_A(a_j) \oplus U_B(a_j)\right)\left(P_A(g) \oplus P_B(g)\right)\right] \cdot \\ &\cdot \left(U_A(a_k) \oplus U_B(a_k)\right)\left(P_A(a) \oplus P_B(a)\right) \|^2 + \\ &+ \| \left(\sqrt{\alpha} \cdot \varphi_A \oplus \sqrt{1-\alpha} \cdot \varphi_B\right) \cdot \prod_{j=1}^{n} \left[\left(U_A(a_j) \oplus U_B(a_j)\right)\left(P_A(g) \oplus P_B(g)\right)\right] \cdot \\ &\cdot \left(U_A(\#) \oplus U_B(\#)\right)\left(P_A(a) \oplus P_B(a)\right) \|^2.\end{aligned}$$

By Equation (1.5), we have

$$\begin{aligned}\mathcal{P}_{C_\oplus}(\omega) = &\sum_{k=1}^{n} \left\| \left[\sqrt{\alpha} \cdot \varphi_A \cdot \prod_{j=1}^{k-1} \left(U_A(a_j)P_A(g)\right) \cdot \left(U_A(a_k)P_A(a)\right)\right] \oplus \right. \\ &\left. \oplus \left[\sqrt{1-\alpha} \cdot \varphi_B \cdot \prod_{j=1}^{k-1} \left(U_B(a_j)P_B(g)\right) \cdot \left(U_B(a_k)P_B(a)\right)\right] \right\|^2 + \\ &+ \left\| \left[\sqrt{\alpha} \cdot \varphi_A \cdot \prod_{j=1}^{n} \left(U_A(a_j)P_A(g)\right) \cdot \left(U_A(\#)P_A(a)\right)\right] \oplus \right. \\ &\left. \oplus \left[\sqrt{1-\alpha} \cdot \varphi_B \cdot \prod_{j=1}^{n} \left(U_B(a_j)P_B(g)\right) \cdot \left(U_B(\#)P_B(a)\right)\right] \right\|^2 \\ = &\alpha \cdot \mathcal{P}_A(\omega) + (1-\alpha) \cdot \mathcal{P}_B(\omega).\end{aligned}$$

∎

While the result of Theorem 5.4.9 is not generally true for MM-QFAs, it holds for the subclass of end-decisive MM-QFAs.

Theorem 5.4.12. *Given two end-decisive* MM-QFA*s A and A', inducing, respectively, the events $\mathcal{P}_A$ and $\mathcal{P}_{A'}$, there always exists an end-decisive* MM-QFA *B such that $\mathcal{P}_B = \mathcal{P}_A \cdot \mathcal{P}_{A'}$.*

Proof. Let

$$A = \langle \varphi, \{M(\sigma)\}_{\sigma \in (\Sigma \cup \{\#\})}, \mathcal{O}_{int}, \mathcal{O}_{fin} \rangle$$
$$A' = \langle \varphi', \{M'(\sigma)\}_{\sigma \in (\Sigma \cup \{\#\})}, \mathcal{O}'_{int}, \mathcal{O}'_{fin} \rangle,$$

where

- $\mathcal{O}_{int} = \{P_{int}(g), P_{int}(r)\}$,
- $\mathcal{O}_{fin} = \{P_{fin}(a), P_{fin}(r)\}$,
- $\mathcal{O}'_{int} = \{P'_{int}(g), P'_{int}(r)\}$,
- $\mathcal{O}'_{fin} = \{P'_{fin}(a), P'_{fin}(r)\}$.

We construct the automaton

$$B = \langle \tilde{\varphi}, \{\tilde{M}(\sigma)\}_{\sigma\in(\Sigma\cup\{\#\})}, \tilde{\mathcal{O}}_{int}, \tilde{\mathcal{O}}_{fin}\rangle,$$

such that

- $\tilde{\varphi} = \varphi \otimes \varphi'$,
- $\tilde{M}(\sigma) = M(\sigma) \otimes M'(\sigma)$,
- $\tilde{\mathcal{O}}_{int} = \{\tilde{P}_{int}(g), \tilde{P}_{int}(r)\}$,
- $\tilde{\mathcal{O}}_{fin} = \{\tilde{P}_{fin}(a), \tilde{P}_{fin}(r)\}$,
- $\tilde{P}_{int}(g) = P_{int}(g) \otimes P'_{int}(g)$,
- $\tilde{P}_{int}(r) = I - \tilde{P}_{int}(g)$,
- $\tilde{P}_{fin}(a) = P_{fin}(a) \otimes P'_{fin}(a)$,
- $\tilde{P}_{fin}(r) = I - \tilde{P}_{fin}(a)$.

Clearly, B is an end-decisive MM-QFA. The event induced by B on input $\omega = a_1a_2\cdots a_n$ is

$$\begin{aligned}
\mathcal{P}_B(\omega) =& || \ (\varphi\otimes\varphi') \prod_{i=1}^{n-1} [(M(a_i)\otimes M'(a_i))\,(P_{int}(g)\otimes P'_{int}(g))] \\
& (M(\#)\otimes M'(\#))(P_{fin}(a)\otimes P'_{fin}(a)) \ ||^2 \\
=& || \ \varphi \prod_{i=1}^{n-1} [M(a_i)P_{int}(g)]\, M(\#)P_{fin}(a)\otimes \\
& \otimes \ \varphi' \prod_{i=1}^{n-1} [M'(a_i)P'_{int}(g)]\, M'(\#)P'_{fin}(a) \ ||^2 \\
=& || \ \varphi \prod_{i=1}^{n-1} [M(a_i)P_{int}(g)]\, M(\#)P_{fin}(a) \ ||^2 \cdot \\
& \cdot || \ \varphi' \prod_{i=1}^{n-1} [M'(a_i)P'_{int}(g)]\, M'(\#)P'_{fin}(a) \ ||^2 \\
=& \mathcal{P}_A(\omega)\cdot\mathcal{P}_{A'}(\omega).
\end{aligned}$$

∎

In what follows, we show that, for every QFA described by vectors and matrices with complex entries, we can always obtain, as described in [20], an equivalent QFA of the same type, by doubling the number of states.

Theorem 5.4.13. *For every m-state* MO-QFA

$$\Omega = \langle \varphi_0,\ \{U(\sigma)\}_{\sigma\in\Sigma},\ P\rangle,$$

where $\varphi_0 \in \mathbb{C}^m$ and $U(\sigma) \in \mathbb{C}^{m\times m}$ for each $\sigma \in \Sigma$, there exists a $2m$-state MO-QFA

$$\Omega' = \langle \varphi'_0,\ \{U'(\sigma)\}_{\sigma\in\Sigma},\ P'\rangle,$$

such that $\varphi'_0 \in \mathbb{R}^{2m}$ and $U(\sigma) \in \mathbb{R}^{2m\times 2m}$ for each $\sigma \in \Sigma$, and $\mathcal{P}_\Omega(\omega) = \mathcal{P}_{\Omega'}(\omega)$, for every $\omega \in \Sigma^*$.

Proof. Let $\{q_j = e_j\}_{1\le j\le m}$ be the set of basis states of Ω and $\{q'_j = e_j\}_{1\le j\le 2m}$ be the set of basis states of Ω' (both represented as characteristic vectors). The idea is to substitute each q_j with the couple q'_{2j-1}, q'_{2j} and decompose on them the complex amplitude z_j associated to q_j, such that $\mathrm{Re}(z_j)$ is associated to q'_{2j-1} and $\mathrm{Im}(z_j)$ is associated to q'_{2j-1}. More formally, we define a function $\phi : \mathbb{C}^n \to \mathbb{R}^{2n}$ mapping the superposition

$$x = \sum_{j=1}^{n} (\alpha_j + i\beta_j)\, q_j$$

into

$$\phi(x) = \sum_{j=1}^{n} \alpha_j q'_{2j-1} + \sum_{j=1}^{n} \beta_j q'_{2j}.$$

With a slight abuse of notation, we use the same function name to denote the mapping on matrices $\phi : \mathbb{C}^{m\times m} \to \mathbb{R}^{2m\times 2m}$ defined in an analogous way: the matrix $\phi(X)$ is obtained by doubling the order of X and substituting the element $(X)_{j,k} = \alpha_{jk} + i\beta_{jk}$ with the matrix

$$\begin{pmatrix} \alpha_{jk} & -\beta_{jk} \\ \beta_{jk} & \alpha_{jk} \end{pmatrix}.$$

It is not hard to see that, for every $v \in \mathbb{C}^n$ and for every couple of complex matrices A and B of order m, the following properties hold

- $\phi(Av) = \phi(A)\phi(v)$,
- $\phi(AB) = \phi(A)\phi(B)$,
- $\phi(A^\dagger) = \phi(A)^T$,

- $v^\dagger v = \phi(v)^T \phi(v)$, i.e., $\|v\| = \|\phi(v)\|$.

We also recall the following definitions:

- A is a unitary matrix if $AA^\dagger = I$
- A is an orthogonal matrix if $AA^T = I$
- A is a complex matrix of orthogonal projection if $A = A^\dagger = A^2$
- A is a real matrix of orthogonal projection if $A = A^T = A^2$.

This implies that the function ϕ maps unitary matrices into orthogonal matrices, and complex matrices of orthogonal projection into real matrices of orthogonal projection. We let

$$\Omega' = \langle \varphi'_0 = \phi(\varphi_0),\ U'(\sigma) = \{\phi(U(\sigma))\}_{\sigma\in\Sigma},\ P' = \phi(P)\rangle.$$

Since the initial configuration φ'_0 satisfies the equivalence

$$\varphi'_0 U'(\omega) P' = \phi(\varphi_0 U(\omega) P)$$

for each $\omega \in \Sigma^*$, it holds $\mathcal{P}_\Omega(\omega) = \mathcal{P}_{\Omega'}(\omega)$. ∎

Theorem 5.4.14. *For every m-state* MM-QFA

$$\Omega = \langle \varphi_0,\ \{U(\sigma)\}_{\sigma\in\Sigma\cup\{\#\}},\ \mathcal{O} = aP(a) + rP(r) + gP(g)\rangle,$$

where $\varphi_0 \in \mathbb{C}^m$ and $U(\sigma) \in \mathbb{C}^{m\times m}$ for each $\sigma \in \Sigma \cup \{\#\}$, there exists a $2m$-state MM-QFA

$$\Omega' = \langle \varphi'_0,\ \{U'(\sigma)\}_{\sigma\in\Sigma\cup\{\#\}},\ \mathcal{O}' = aP'(a) + rP'(r) + gP'(g)\rangle,$$

such that $\varphi'_0 \in \mathbb{R}^{2m}$ and $U(\sigma) \in \mathbb{R}^{2m\times 2m}$ for each $\sigma \in \Sigma \cup \{\#\}$, and $\mathcal{P}_\Omega(\omega) = \mathcal{P}_{\Omega'}(\omega)$, for every $\omega \in \Sigma^*$.

Proof. The proof is analogous to the Measure-Once case: by letting ϕ be the function defined in the proof of Theorem 5.4.13, we set

- $\varphi'_0 = \phi(\varphi_0)$,
- $U'(\sigma) = \phi(U(\sigma))$, for each $\sigma \in \Sigma \cup \{\#\}$,
- $P'(x) = \phi(P(x))$, for $x \in \{a, r, g\}$,

so that, for every word $\omega = \sigma_1\sigma_2\cdots\sigma_n$, by letting $\sigma_{n+1} = \#$, it holds

$$\varphi_0' \left(\prod_{j=1}^{k-1} (U'(\sigma_j)P'(g)) \right) U'(\sigma_k)P'(a) = \phi \left(\varphi_0 \left(\prod_{j=1}^{k-1} (U(\sigma_j)P(g)) \right) U(\sigma_k)P(a) \right),$$

for each $1 \leq k \leq n+1$. ∎

Theorem 5.4.15. *For every (m,k)-state* QFC

$$\Omega = \langle \varphi_0,\ \{U(\sigma)\}_{\sigma\in\Sigma\cup\{\#\}},\ \mathcal{O} = \sum_{c_j\in C} c_j P(c_j),\ \mathcal{L} \rangle,$$

where $\varphi_0 \in \mathbb{C}^m$ and $U(\sigma) \in \mathbb{C}^{m\times m}$ for each $\sigma \in \Sigma\cup\{\#\}$, there exists a $(2m,k)$-state QFC

$$\Omega' = \langle \varphi_0',\ \{U'(\sigma)\}_{\sigma\in\Sigma\cup\{\#\}},\ \mathcal{O}' = \sum_{c_j\in C} c_j P'(c_j),\ \mathcal{L} \rangle,$$

such that $\varphi_0' \in \mathbb{R}^{2m}$ and $U(\sigma) \in \mathbb{R}^{2m\times 2m}$ for each $\sigma \in \Sigma\cup\{\#\}$, and $\mathcal{P}_\Omega(\omega) = \mathcal{P}_{\Omega'}(\omega)$, for every $\omega \in \Sigma^*$.

Proof. The proof is analogous to the Measure-Once case: by letting ϕ be the function defined in the proof of Theorem 5.4.13, we set

- $\varphi_0' = \phi(\varphi_0)$,
- $U'(\sigma) = \phi(U(\sigma))$, for each $\sigma \in \Sigma\cup\{\#\}$,
- $P'(c_j) = \phi(P(c_j))$, for every $c_j \in C$,

so that, for every couple of words $x_1x_2\cdots x_n \in (\Sigma\cup\{\#\})^*$ and $y_1y_2\cdots y_n \in C^*$, it holds

$$\varphi_0' \prod_{j=1}^{n} (U'(x_j)P'(y_j)) = \phi \left(\varphi_0 \prod_{j=1}^{n} (U(x_j)P(y_j)) \right).$$

∎

5.5 Characterization of unary languages recognized by MM-QFAs

In [2], it is stated that MM-QFAs recognize any regular language corresponding to **EJ**, which is the variety of monoids whose idempotents generate a J-trivial monoid, where J is the Green's relation determined by two-sided ideals. Since any syntactic monoid of a unary regular language belongs to **EJ**, the results in [2] imply that MM-QFAs recognize any unary regular language.

In this section, we constructively prove that any unary regular language can be accepted by a MM-QFA with constant cut point and isolation. Since, as recalled in Section 5.4, MM-QFAs with isolated cut point accept regular languages only, this provides a characterization of the recognition power of MM-QFAs in the unary case.

We remark that, by Theorem 2.4.1, unary regular languages are ultimately periodic sets. It is well known that unary MO-QFAs with isolated cut point are able to accept periodic languages only [7, 21, 45]. So, we will focus here on building unary MM-QFAs having both transient and ergodic components. For conciseness of notation, in what follows we describe an $m \times m$ observable $\mathcal{O} = a \cdot P(a) + r \cdot P(r) + g \cdot P(g)$ by a vector $v \in \{a, r, g\}^m$, where $v_j = x$ if and only if $(P(x))_{j,j} = 1$, for $x \in \{a, r, g\}$.

Theorem 5.5.1. *Let $L \subseteq \sigma^*$ be a unary language recognized by a* DFA *with $T+1$ transient states and P ergodic states. Then, there exists a* MM-QFA *with $2(T+2)+P$ basis states recognizing L with cut point $\frac{3}{8}$ isolated by $\frac{1}{8}$.*

Proof. Since the DFA for L has $T+1$ transient and P ergodic states, L can be clearly regarded as the disjoint union of two languages: the finite language $L_T = L \cap \{\sigma\}^{\leq T}$, and the ultimately periodic language $L_P = L \cap \{\sigma\}^{>T}$ of period P. The idea of this construction is to build two different QFAs A_T and A_{P°, such that A_T mimics the DFA for L_T, while A_{P° recognizes a cyclic language that coincides with L_P on words longer than T. Then we combine them with a third QFA $A_{\hat{T}}$, which "activates" A_{P° only after a portion of length T of the input has been read.

We start by establishing that the language L_T is accepted by the end-decisive MM-QFA

$$A_T = \langle \varphi_0, \{U(\sigma), U(\#)\}, \mathcal{O}_{int}, \mathcal{O}_{fin} \rangle$$

with $T+2$ basis states, where

- $\varphi_0 = e_1 = (1, 0, \ldots, 0) \in \mathbb{C}^{T+2}$,
- $U(\sigma) = U(\#) = \begin{pmatrix} 0 & 1 & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & & & 1 \\ 1 & 0 & \ldots & 0 \end{pmatrix} \in \mathbb{C}^{(T+2)\times(T+2)}$ is the right circular permutation matrix,
- $\mathcal{O}_{int} = (r, g, \ldots, g)$,
  $\mathcal{O}_{fin} = (r, x_0, \ldots, x_T)$, with $x_i = \begin{cases} a & \text{if } \sigma^i \in L_T \\ r & \text{otherwise.} \end{cases}$

Notice that the state vector of A_T, of length $T+2$, mimics the state vector of the DFA for L_T, which has $T+1$ states. The additional component in A_T is due to the end-marker #. In fact, we show that A_T has actually a deterministic behavior:

$$U(\sigma)P_{int}(g) = \begin{pmatrix} 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \\ 0 & & & 1 \\ 0 & 0 & \dots & 0 \end{pmatrix},$$

$$\left(U(\sigma)P_{int}(g)\right)^2 = \begin{pmatrix} 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & 1 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

$$\vdots$$

$$\left(U(\sigma)P_{int}(g)\right)^T = \begin{pmatrix} 0 & \dots & 1 & 0 \\ 0 & & 0 & 1 \\ \vdots & \ddots & & \vdots \\ 0 & \dots & 0 & 0 \end{pmatrix},$$

In other words, applying the matrix $(U(\sigma)P_{int}(g))^j$ to the current state vector has the effect of performing j times the right circular shift, and then deleting the first j components, which implies that, for $j \geq T+2$, $(U(\sigma)P_{int}(g))^j$ is the null matrix.

Therefore, the event induced by A_T writes as

$$\mathcal{P}_{A_T}(\sigma^n) = \left\|\varphi_0 \left(U(\sigma)P_{int}(g)\right)^n U(\#)P_{fin}(a)\right\|^2 = \left\|\xi P_{fin}(a)\right\|^2,$$

where

$$\xi = \begin{cases} e_{n+2} & \text{if } n \leq T \\ e_1 & \text{if } n = T+1 \\ \mathbf{0} & \text{if } n > T+1, \end{cases}$$

which implies

$$\mathcal{P}_{A_T}(\sigma^n) = \begin{cases} 1 & \text{if } \sigma^n \in L_T \\ 0 & \text{if } \sigma^n \notin L_T. \end{cases}$$

We define now our second MM-QFA, which is actually a MO-QFA, with P basis states, accepting a language L_{P° such that $L_{P^\circ} \cap \{\sigma\}^{>T} = L_P$. We have

$$A_{P^\circ} = \langle \varphi_0, \{U(\sigma), U(\#)\}, \mathcal{O}_{int}, \mathcal{O}_{fin} \rangle$$

where

- $\varphi_0 = e_1 = (1, 0, \ldots, 0)$,
- $U(\sigma) = \begin{pmatrix} 0 & 1 & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & & & 1 \\ 1 & 0 & \ldots & 0 \end{pmatrix} \in \mathbb{C}^{(P)\times(P)}$ is the right circular permutation matrix,
  $U(\#) = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \ldots & 1 \end{pmatrix} \in \mathbb{C}^{(P)\times(P)}$ is the identity matrix,
- $\mathcal{O}_{int} = (g, \ldots, g)$,
  $\mathcal{O}_{fin} = (x_0, \ldots, x_{P-1})$, with $x_j = \begin{cases} a & \text{if } \sigma^{P\lceil (T+1)/P\rceil + j} \in L_P \\ r & \text{otherwise.} \end{cases}$

It is easy to see that $L_{P^\circ} = \left\{\sigma^{j+kP} \mid k \in \mathbb{N}, \sigma^{P\lceil (T+1)/P\rceil + j} \in L\right\}$, which is a language coinciding with L on the strings of length exceeding T.

Finally, we need a third MM-QFA

$$A_{\widehat{T}} = \langle \varphi_0', \{U'(\sigma), U'(\#)\}, \mathcal{O}_{int}', \mathcal{O}_{fin}' \rangle,$$

with $T + 2$ basis states, accepting the language $\{\sigma\}^{>T}$. The automaton $A_{\widehat{T}}$ is defined as A_T, except for the observables which are now $\mathcal{O}_{int}' = (a, g, \ldots, g)$ and $\mathcal{O}_{fin}' = (a, r, \ldots, r)$. Analyzing the behavior of $A_{\widehat{T}}$ we have that

- if $n \leq T$:

$$\begin{aligned} \mathcal{P}_{\widehat{T}}(\sigma^n) =& \sum_{k=0}^{n-1} \left\| e_1 \left(U'(\sigma) P_{int}'(g)\right)^k U'(\sigma) P_{int}'(a) \right\|^2 \\ &+ \left\| e_1 \left(U'(\sigma) P_{int}'(g)\right)^n U'(\#) P_{fin}'(a) \right\|^2 \\ =& \sum_{k=0}^{n-1} \underbrace{\left\| e_{k+2} P_{int}'(a) \right\|^2}_{0} + \underbrace{\left\| e_{n+2} P_{fin}'(a) \right\|^2}_{0} = 0, \end{aligned}$$

- if $n = T + 1$:

$$\begin{aligned} \mathcal{P}_{\widehat{T}}(\sigma^n) =& \sum_{k=0}^{T} \left\| e_1 \left(U'(\sigma) P_{int}'(g)\right)^k U'(\sigma) P_{int}'(a) \right\|^2 \\ &+ \left\| e_1 \left(U'(\sigma) P_{int}'(g)\right)^{T+1} U'(\#) P_{fin}'(a) \right\|^2 \\ =& 0 + \left\| e_1 P_{fin}'(a) \right\|^2 = 1, \end{aligned}$$

- if $n > T+1$:

$$\begin{aligned}\mathcal{P}_{\widehat{T}}(\sigma^n) =& \sum_{k=0}^{T} \left\| e_1 \left(U'(\sigma)P'_{int}(g)\right)^k U'(\sigma)P'_{int}(a) \right\|^2 \\ &+ \left\| e_1 \left(U'(\sigma)P'_{int}(g)\right)^{T+1} U'(\sigma)P'_{int}(a) \right\|^2 \\ &+ \sum_{k=T+2}^{n} \left\| e_1 \left(U'(\sigma)P'_{int}(g)\right)^k U'(\sigma)P'_{int}(a) \right\|^2 \\ &+ \left\| e_1 \left(U'(\sigma)P'_{int}(g)\right)^n U'(\#)P'_{fin}(a) \right\|^2 \\ =& 0 + \left\| e_1 P'_{int}(a) \right\|^2 + 0 + 0 = 1.\end{aligned}$$

Since, as recalled in Theorem 5.4.11, the class of events induced by MM-QFAs is closed under convex linear combination, it is possible to construct a MM-QFA A_L with $2(T+2)+P$ basis states inducing the event

$$\mathcal{P}_{A_L}(\sigma^n) = \frac{1}{2}\mathcal{P}_{A_T}(\sigma^n) + \frac{1}{4}\mathcal{P}_{A_{P^\circ}}(\sigma^n) + \frac{1}{4}\mathcal{P}_{A_{\widehat{T}}}(\sigma^n).$$

Notice that the MM-QFAs A_T, A_{P° and $A_{\widehat{T}}$ induce events in $\{0,1\}$. Moreover, when $\sigma^n \notin L$, the event $\mathcal{P}_{A_L}(\sigma^n)$ can only assume values 0 or 1/4, while for $\sigma^n \notin L$, $\mathcal{P}_{A_L}(\sigma^n)$ can be either 1/2, 3/4 or 1. ∎

Chapter 6

Descriptional complexity of quantum automata

In the literature, there are many results on the descriptional complexity of QFAs: in several cases, MO-QFAs turn out to be more succinct than classical counterparts. As a typical example, for a fixed prime n, consider the unary language $L_n = \{\sigma^{kn} \mid k \in \mathbb{N}\}$. For L_n, n states are necessary and sufficient on isolated cut point probabilistic automata [41]. On the other hand, in [12], a MO-QFA for L_n is exhibited with isolation $(1-\varepsilon)/2$ and $O(\log n/\varepsilon^3)$ states. Several other results on the descriptional complexity of QFAs can be found, e.g., in [3, 10, 30]. In [10], probabilistic techniques are proposed for constructing small size MO-QFAs working on a non-unary alphabet Σ inducing periodic stochastic events (in this case, a different periodicity for each symbol in Σ occurring in the input word is considered). This has lead to a MO-QFA with $O(|\Sigma| \log n)$ states recognizing with isolated cut point the language $L \subseteq \Sigma^*$ consisting of the strings in which the number of occurrences of each symbol in Σ is a multiple of n. Notice that $n^{|\Sigma|}$ states are necessary and sufficient for recognizing L on a deterministic automaton. In [40], the techniques used in [10] have been extended to construct small size MO-QFAs inducing multiperiodic events defined as boolean combinations of periodicity constraints on the occurrences of the symbols in the input word.

In this Chapter, we analyze the succinctness of the Measure-Many model. In particular, we improve, for some families of unary languages, the construction presented in the proof of Theorem 5.5.1, reducing exponentially the size of the minimal MM-QFA. To this aim, we adapt techniques known for MO-QFAs, to induce ε-approximations of periodic events. We then complete these results, which were published in [16, 17], by studying the maximal gap in the descriptional complexity between DFAs and the quantum model. We analyze the more general case of QFCs working on alphabets of arbitrary size, and we prove a state complexity

lower bound in the conversion from QFC to DFA, which is very close to the exponential gap reached in the preceding Section. In order to obtain this conversion, we adapt a technique first introduced by Rabin in [49] for the conversion from PFAs to DFAs, and proposed again in [12] for converting MO-QFAs into DFAs.

For simplicity, when we deal with unary QFAs, we define the induced event on the set $\mathbb{N}$ instead of Σ^*, i.e., for an automaton Ω, we consider the induced event $p_\Omega : \mathbb{N} \to [0,1]$ such that, for any $n \in \mathbb{N}$,

$$p_\Omega(n) = \mathcal{P}_\Omega(\sigma^n).$$

We here recall some definitions and results about events:

Definition An event $p : \mathbb{N} \to [0,1]$ is called *d-periodic* if, for any $k \geq 0$, it holds

$$p(k) = p(k+d).$$

This implies that a d-periodic event is completely described by the sequence $\{p(j)\}_{0 \leq j \leq d-1}$, i.e., it can be represented by the vector

$$(p(0), p(1), \ldots, p(d-1)).$$

Definition Given a d-periodic event $p : \mathbb{N} \to [0,1]$, its *discrete Fourier transform* is the complex vector $P = (P(0), P(1), \ldots, P(d-1))$ such that

$$P(j) = \sum_{k=0}^{d-1} p(k) \cdot e^{i\frac{2\pi}{d}kj}. \tag{6.1}$$

The event p is completely described by the vector P, in fact it can be obtained from P by the inverse formula

$$p(k) = \frac{1}{d} \sum_{j=0}^{d-1} P(j) \cdot e^{-i\frac{2\pi}{d}kj}. \tag{6.2}$$

We call $\mathcal{F}(p)$ the discrete Fourier transform of an event p.

Definition Given the events $p : \mathbb{N} \to [0,1]$, $q : \mathbb{N} \to [0,1]$ and a positive value $\varepsilon \in \mathbb{R}$, the event q is an *ε-approximation* of p if it holds

$$\sup_{n \in \mathbb{N}} \{|p(n) - q(n)|\} \leq \varepsilon.$$

In what follows, we let $f_\varepsilon(n)$ denote any function satisfying, for each n, the inequality $|f_\varepsilon(n)| \leq \varepsilon$. We say that a QFA Ω *ε-approximates* an event p if p_Ω is a ε-approximation of p.

We now present a condition on the Fourier transform of a d-periodic event p, which allows us to obtain a QFA, of size logarithmic in the period d, which approximates p.

Theorem 6.0.2. *[11] Given a d-periodic event p, the event*

$$\frac{1}{2}+\frac{1}{2}\frac{d}{\|\mathcal{F}(p)\|_1}p$$

is ε-approximated by a MO-QFA *with $O\left(\frac{\log d}{\varepsilon^2}\right)$ states.*

Corollary 6.0.3. *If $\|\mathcal{F}(p)\|_1 \leq d$, then the event $\frac{1}{2}+\frac{1}{2}p$ is ε-approximated by a* MO-QFA *with $O\left(\frac{\log d}{\varepsilon^2}\right)$ states.*

Using this result, we can improve our construction in Theorem 5.5.1 from a descriptional complexity point of view. In fact, we prove the following

Theorem 6.0.4. *Let $L \subseteq \sigma^*$ be a language recognized by a* DFA *with $T+1$ transient and P ergodic states, and let $L_{P^\circ} = \left\{\sigma^{i+kP} \mid k \in \mathbb{N}, \sigma^{P\lceil (T+1)/P\rceil+i} \in L\right\}$, with χ_{P° being its characteristic function. If $\|\mathcal{F}(\chi_{P^\circ})\|_1 \leq P$, then L can be recognized by a* MM-QFA *with cut point $\frac{7}{16}$ isolated by a constant $\delta < \frac{1}{16}$, and $O\Big(T+\frac{\log(P)}{(1-16\delta)^2}\Big)$ basis states.*

Proof. Let us consider the MM-QFA A_L given in the proof of Theorem 5.5.1, and focus on its component A_{P°. We notice that A_{P° is a MO-QFA which recognizes L_{P° by inducing the P-periodic event $p_{A_{P^\circ}} = \chi_{P^\circ}$. By Corollary 6.0.3, if $\|\mathcal{F}(p_{A_{P^\circ}})\|_1 \leq P$ holds, we can induce the event

$$p(n) = \frac{1}{2}p_{A_T}(n) + \frac{1}{4}\left(\frac{1}{2}+\frac{1}{2}p_{A_{P^\circ}}(n) + f_\varepsilon(n)\right) + \frac{1}{4}p_{A_{\tilde{T}}}(n)$$

by a MM-QFA with $O(T+\frac{\log P}{\varepsilon^2})$ basis states. For every $n \in \mathbb{N}$, we have that $\sigma^n \in L$ implies $p(n) \geq \frac{1}{2}-\frac{\varepsilon}{4}$, while $\sigma^n \notin L$ implies $p(n) \leq \frac{3}{8}+\frac{\varepsilon}{4}$. We require $\frac{1}{2}-\frac{\varepsilon}{4} = \lambda+\delta$ and $\frac{3}{8}+\frac{\varepsilon}{4} = \lambda-\delta$. So, we get $\lambda = \frac{7}{16}$ and $\delta = \frac{1}{16}-\frac{\varepsilon}{4}$, and the result follows. ∎

6.1 MM-QFAs exponentially smaller than DFAs

In what follows, we single out unary languages accepted by MM-QFAs having a number of states logarithmic in both the parameters T and P. To this aim, it will be useful to consider the end-decisive MM-QFA

$$A_\theta = \langle \varphi_0, \{U(\sigma), U(\#)\}, \mathcal{O}_{int}, \mathcal{O}_{fin}\rangle$$

with 2 basis states, where

- $\varphi_0 = (1,0)$,

- $U(\sigma) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$,
- $U(\#) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$,
- $\mathcal{O}_{int} = (g, r)$,
- $\mathcal{O}_{fin} = (a, a)$.

We have that

$$p_{A_\theta}(n) = \left\| \varphi_0 \left(U(\sigma) \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right)^n \left(U(\#) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \right\|^2 = \\ = \|((\cos(\theta))^n, 0)\|^2 = (\cos(\theta))^{2n}.$$

The first family we consider consist of unary languages of the form $L_T \cup L_P$, where L_T is a finite language recognized by a DFA with $T+1$ transient states (without loss of generality, we assume $T > 1$) and one ergodic state (which is the trap state), and L_P is a periodic language of period P. Let χ_{T° and χ_P be the characteristic functions of the languages $L_{T^\circ} = L_T \cdot \{\sigma^{T^3}\}^*$ and L_P, respectively. Then, the following holds:

Theorem 6.1.1. *If $\|\mathcal{F}(\chi_{T^\circ})\|_1 \leq T^3$ and $\|\mathcal{F}(\chi_P)\|_1 \leq P$, then the language $L_T \cup L_P$ can be recognized by a* MM-QFA *with isolated cut point and $O(\log(N))$ basis states, where $N = \max\{T, P\}$.*

Proof. By Corollary 6.0.3, if $\|\mathcal{F}(\chi_{T^\circ})\|_1 \leq T^3$, there exists a MO-QFA A_{T° with $O(\frac{\log T}{\varepsilon^2})$ basis states inducing an ε-approximation of the event $\frac{1}{2} + \frac{1}{2}\chi_{T^\circ}$. Note that A_{T° (as well as all MO-QFAs) is an end-decisive MM-QFA. Therefore, by applying Theorem 5.4.12 to A_{T° and A_θ with $\theta = \arccos\left(\left(\frac{6}{7}\right)^{\frac{1}{2T}}\right)$, we get a MM-QFA with $O(\frac{\log T}{\varepsilon^2})$ basis states inducing the event $\left(\frac{6}{7}\right)^{\frac{n}{T}} \left(\frac{1}{2} + \frac{1}{2}\chi_{T^\circ}(n) + f_\varepsilon(n)\right)$. By letting $\alpha = \frac{3}{8}$ and $\varepsilon = \frac{1}{32}$, the event

$$\psi(n) = \alpha \left(\frac{6}{7}\right)^{\frac{n}{T}} \left(\frac{1}{2} + \frac{1}{2}\chi_{T^\circ}(n) + f_\varepsilon(n)\right) + (1-\alpha)\left(\frac{1}{2} + \frac{1}{2}\chi_P(n) + f_\varepsilon(n)\right) \quad (6.3)$$

can be induced by a MM-QFA with $O(\log T + \log P)$ states. We claim that this MM-QFA recognizes $L = L_T \cup L_P$ with isolated cut point. Indeed, let the cut point $\lambda = \frac{9}{16}$ and isolation $\delta = \frac{1}{64}$. If $\sigma^n \in L$, we have:

- $\sigma^n \in L_T$ implies $\psi(n) \geq \frac{3}{8} \cdot \frac{6}{7}(1-\varepsilon) + \frac{5}{8}(\frac{1}{2} - \varepsilon) \geq \lambda + \delta$,
- $\sigma^n \in L_P$ implies $\psi(n) \geq \frac{5}{8}(1-\varepsilon) \geq \lambda + \delta$.

For $\sigma^n \notin L$, we have:

- $n \leq T^3$ implies $\psi(n) \leq \frac{1}{2} + \varepsilon \leq \lambda - \delta$,
- $n > T^3$ implies $\psi(n) \leq \frac{3}{8}(\frac{6}{7})^{T^2} + \frac{5}{8}(\frac{1}{2} + \varepsilon) \leq \lambda - \delta$.

∎

Notice that finite unary languages can be seen as languages in the form $L_T \cup L_P$, with $L_P = \emptyset$. Theorem 6.1.1 therefore enables us to show the existence of finite languages that can be recognized by MM-QFAs exponentially smaller than the corresponding DFAs. More precisely, we get

Corollary 6.1.2. *If $\|\mathcal{F}(\chi_{T^\circ})\|_1 \leq T^3$, then there exists a* MM-QFA *recognizing L_T with isolated cut point and $O(\log(T))$ basis states.*

Proof. For a finite language L_T, we can induce the event $\psi(n)$ described in Equation (6.3) with $O(\log(T))$ basis states, choosing $\alpha = 1$ in order to have a cut point with a better isolation. Let us fix the approximation rate $\varepsilon = \frac{1}{8}$, cut point $\lambda = \frac{11}{16}$, and isolation $\delta = \frac{1}{16}$. For $0 \leq n \leq T^3$, we have:

- $\sigma^n \in L_T$ implies $\psi(n) \geq \frac{6}{7}(1 - \varepsilon) = \lambda + \delta$,
- $\sigma^n \notin L_T$ implies $\psi(n) \leq \frac{1}{2} + \varepsilon = \lambda - \delta$.

For $n > T^3$, we have $\sigma^n \notin L_T$ and $\psi(n) < \left(\frac{6}{7}\right)^{T^2} < \lambda - \delta$. ∎

We can extend our result on finite languages to the other family of ultimately periodic languages of period 1, i.e., languages of the form $L_{T'} = L_T \cup \{\sigma\}^{>T}$, where $L_T \subseteq \{\sigma\}^{\leq T}$. Clearly, $L_{T'}$ is recognized by a DFA with $T + 1$ transient and 1 ergodic states, where the only difference from the DFA for L_T is that now the ergodic state is accepting. Denote the complement of $L_{T'}$ by $L^c_{T'}$, and let $\bar{\chi}_{T^\circ}$ be the characteristic function of $L^c_{T'} \cdot \{\sigma^{T^3}\}^*$. For the language $L_{T'}$, we have

Theorem 6.1.3. *If $\|\mathcal{F}(\bar{\chi}_{T^\circ})\|_1 \leq T^3$, then there exists a* MM-QFA *recognizing $L_T \cup \{\sigma\}^{>T}$ with isolated cut point and $O(\log(T))$ basis states.*

Proof. By arguments similar to those used for Corollary 6.1.2, and by the closure properties of the events induced by MM-QFAs, we can induce the event

$$\phi(n) = 1 - \left(\frac{6}{7}\right)^{\frac{n}{T}} \left(\frac{1}{2} + \frac{1}{2}\bar{\chi}_{T^\circ}(n) + f_\varepsilon(n)\right)$$

with $O(\log(T))$ basis states. Hence, the language $L_T \cup \{\sigma\}^{>T}$ can be recognized with cut point $\frac{5}{16}$ isolated by $\frac{1}{16}$. ∎

6.2 Bound on the conversion cost from QFCs to DFAs

The descriptional complexity results of the previous section show that it is possible to achieve an exponential gap between the size of MM-QFAs and the one of DFAs. Now we wonder if that is the best achievable result, i.e., we look for an upper bound in the conversion from a quantum automaton to a deterministic device. To this aim, we study the most general case of QFCs recognizing languages over an alphabet Σ of arbitrary size. We divide the study of this conversion into three parts: first, we describe a matrix representation for QFCs, introduced in [10] by Bertoni, Mereghetti, Palano, and we analyze the properties of such representation, then we construct the equivalent DFA by adapting Rabin's technique presented in [49]. Finally, we analyze the state complexity of the resulting DFA with respect to the size of the original QFC.

Linear representation of QFCs:

Let $\Gamma = \Sigma \cup \{\#\}$ and let

$$\Omega = \langle \phi,\ \{U(\sigma)\}_{\sigma\in\Gamma},\ \mathcal{O} = \sum_{c\in C} cP(c),\ \mathcal{L}\rangle$$

be a QFC, with δ-isolated cut point λ, and let

$$D = \langle \alpha,\ \{M(c)\}_{c\in C},\ \beta\rangle$$

be the minimal DFA realizing the characteristic function $\chi_{\mathcal{L}}$ of $\mathcal{L}$. Denote by q and k the number of quantum and classical states of Ω. By Theorem 5.4.15, there exists a QFC

$$\Omega_R = \langle \phi_R,\ \{U_R(\sigma)\}_{\sigma\in\Gamma},\ \mathcal{O}_R = \sum_{c\in C} cP_R(c),\ \mathcal{L}\rangle$$

equivalent to Ω with $2q$ quantum states and k classical states, such that ϕ_R and U_R have real entries.

We define the linear form

$$\langle \varphi_0,\ \{V(\sigma)\}_{\sigma\in\Gamma},\ \eta\rangle,$$

such that

- $\varphi_0 = (\phi_R \otimes \phi_R \otimes \alpha)$,
- $V(\sigma) = (U_R(\sigma) \otimes U_R(\sigma) \otimes I) \cdot \sum_{c\in C} P(c) \otimes P(c) \otimes M(c)$,

- $\eta = \sum_{k=1}^{2q} e_k \otimes e_k \otimes \beta$.

As shown in [10], for any word $\omega \in \Sigma^*$ it holds $\|\varphi_0 V(\omega)\| \leq 1$, so all the quantum states in this representation are vectors in the unitary ball $\mathcal{B}_1(0)$ of dimension $4q^2k$, which we call here $\mathcal{B}_1$ for brevity. Moreover, the above linear form is an alternative representation for the automaton Ω:

Theorem 6.2.1. *[10] For any $\omega = \sigma_1\sigma_2\cdots\sigma_n \in \Gamma^*$, it holds*

$$p_\Omega(\omega) = \varphi_0 V(\omega)\eta.$$

Proof. From Theorem 5.4.15 we know that $p_\Omega(\omega) = p_{\Omega_R}(\omega)$, we need to show that $p_{\Omega_R}(\omega) = \varphi_0 V(\omega)\eta$. By definition, we have

$$\begin{aligned}\varphi_0 V(\omega)\eta =& (\phi_R \otimes \phi_R \otimes \alpha)\prod_{j=1}^{n}\left[(U_R(\sigma_j)\otimes U_R(\sigma_j)\otimes I)\cdot\sum_{c\in C} P(c)\otimes P(c)\otimes M(c)\right]\cdot \\ &\cdot\left(\sum_{k=1}^{2q} e_k\otimes e_k\otimes\beta\right) \\ =&\sum_{k=1}^{m}\sum_{y=y_1\cdots y_n\in C^n}\left|\left(\phi_R\prod_{j=1}^{n}U_R(\sigma_j)P(y_j)\right)_k\right|^2\cdot\alpha M(y)\beta,\end{aligned}$$

where each y is a word in C^n, i.e., each y_j represents a possible result of the observation after evolving on the symbol σ_j. The factor $\alpha M(y)\beta$ has the effect of selecting only the words y belonging to the control language $\mathcal{L}$, so the above equation can be rewritten as

$$\begin{aligned}\varphi_0 V(\omega)\eta &= \sum_{y=y_1\cdots y_n\in C^n}\chi_{\mathcal{L}}(y)\sum_{k=1}^{m}\left|\left(\phi_R\prod_{j=1}^{n}U_R(\sigma_j)P(y_j)\right)_k\right|^2 \\ &= \sum_{y=y_1\cdots y_n\in\mathcal{L}}\left\|\phi_R\prod_{j=1}^{n}U_R(\sigma_j)P(y_j)\right\|^2 = p_{\Omega_R}(\omega).\end{aligned}$$

∎

We call $\langle\varphi_0, \{V(\sigma)\}_{\sigma\in\Gamma}, \eta\rangle$ the *real-linear representation* of Ω.

The following result was also shown in [10]:

Lemma 6.2.2. *Let $U(\sigma)$ be a unitary matrix, for $\sigma \in \Sigma$ and $\mathcal{O}$ an observable with results in C described by projectors $P(c)$, for $c \in C$. For any complex vector φ and word $\sigma_1\cdots\sigma_n \in \Sigma^n$, we get*

$$\sum_{y=y_1\cdots y_n\in C^n}\|\varphi\prod_{j=1}^{n}U(\sigma_j)P(y_j)\|^2 = \|\varphi\|^2.$$

The next two technical Lemmas will be useful in the rest of the section:

Lemma 6.2.3. *For any couple of vectors $v, v' \in \mathcal{B}_1$ such that $\|v'\| \geq \|v\|$ and $\cos(\text{ang}(v', v)) \geq 0$, it holds*

$$\|v' - v\| \frac{\|v'\|}{\|v\|} \geq \left\| v' - \frac{\|v'\|}{\|v\|} v \right\|.$$

Proof. For the sake of readability, let $r = \frac{\|v'\|}{\|v\|}$. The property holds if

$$\|v' - v\|^2 r^2 \geq \|v' - rv\|^2,$$

therefore we have

$$\begin{aligned} r^2\|v'\|^2 + r^2\|v\|^2 - 2r^2\langle v', v\rangle \geq & \|v'\|^2 + r^2\|v\|^2 - 2r\langle v', v\rangle \\ & \Updownarrow \\ (r^2-1)\|v'\|^2 - 2(r^2 - r)\langle v', v\rangle \geq & 0. \end{aligned}$$

Since $r \geq 1$, $\|v'\| = r\|v\|$ and $\cos(\text{ang}(v', v)) \geq 0$, we have

$$(r^2-1)\|v'\|^2 - 2(r^2 - r)\langle v', v\rangle \geq (r^2-1)\|v'\|^2 - 2(r-1)\|v'\|^2 = (r-1)^2\|v'\|^2,$$

which is always positive, so the Lemma holds. ∎

Lemma 6.2.4. *Given two vectors $v, v' \in \mathcal{B}_1$ such that*

(i) $\|v'\| \geq \|v\|$, *and*

(ii) $\cos(\text{ang}(v', v)) \geq 0$,

by calling $\theta = \text{ang}(v' - v, v)$, it holds that

$$\cos(\theta) \geq -\frac{1}{\sqrt{2}}.$$

Proof. This property can be intuitively visualized by considering the 2-dimensional section of $\mathcal{B}_1$ containing the origin, v and v', shown in Figure 6.1.

In order to prove this mathematically, we first show that the worst case happens when v and v' have the same length, i.e. we claim that, by calling $\theta' = \text{ang}(v' \frac{\|v\|}{\|v'\|} - v, v)$, it holds

$$\cos(\theta) \geq \cos(\theta'). \tag{6.4}$$

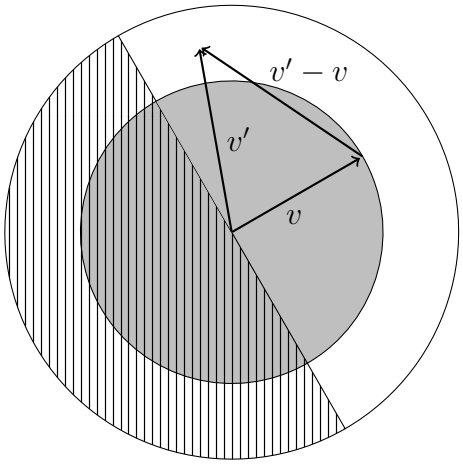


Figure 6.1: 2-dimensional section of $\mathcal{B}_1$ containing the origin, v and v'. The vector v' can only be in the white part: the gray part is forbidden by the constraint $\|v'\| \geq \|v\|$, while the part with the vertical lines is forbidden by the constraint $\cos(\text{ang}(v', v)) \geq 0$. Clearly, the worst case is when $\|v'\| = \|v\|$ and the angle between them is $\frac{\pi}{4}$: in that case, we have $\theta = \frac{3\pi}{4}$.

Clearly this holds if $\|v\|$=0, because in this case $\cos(\theta) = \cos(\theta') = 0$. If $\|v\| > 0$

we have

$$\begin{aligned}
\cos(\theta) \geq & \cos(\theta') \\
& \Updownarrow \\
\frac{\langle v'-v, v\rangle}{\|v'-v\|\|v\|} \geq & \frac{\left\langle \frac{\|v\|}{\|v'\|}v'-v, v\right\rangle}{\left\|\frac{\|v\|}{\|v'\|}v'-v\right\|\|v\|} \\
& \Updownarrow \\
\frac{\langle v', v\rangle - \|v\|^2}{\|v'-v\|\|v\|} \geq & \frac{\left\langle \frac{\|v\|}{\|v'\|}v', v\right\rangle - \|v\|^2}{\left\|\frac{\|v\|}{\|v'\|}v'-v\right\|\|v\|} \\
& \Updownarrow \\
\frac{\|v\|^2 - \langle v', v\rangle}{\|v'-v\|\|v\|} \leq & \frac{\|v\|^2 - \frac{\|v\|}{\|v'\|}\langle v', v\rangle}{\left\|v'-\frac{\|v'\|}{\|v\|}v\right\|\frac{\|v\|^2}{\|v'\|}} \\
& \Updownarrow \\
\frac{\|v\|^2 - \langle v', v\rangle}{\|v'-v\|} \leq & \frac{\|v\|\|v'\| - \langle v', v\rangle}{\left\|v'-\frac{\|v'\|}{\|v\|}v\right\|} \\
& \Updownarrow \\
\frac{\|v\|\|v'\| - \frac{\|v'\|}{\|v\|}\langle v', v\rangle}{\|v'-v\|\frac{\|v'\|}{\|v\|}} \leq & \frac{\|v\|\|v'\| - \langle v', v\rangle}{\left\|v'-\frac{\|v'\|}{\|v\|}v\right\|}.
\end{aligned} \tag{6.5}$$

Since $\|v'\| \geq \|v\|$, it holds $\|v\|\|v'\| - \langle v', v\rangle \geq \|v\|\|v'\| - \frac{\|v'\|}{\|v\|}\langle v', v\rangle$ and, since $\|v\|\|v'\| - \langle v', v\rangle \geq 0$, Equation (6.5) is verified if $\|v'-v\|\frac{\|v'\|}{\|v\|} \geq \left\|v'-\frac{\|v'\|}{\|v\|}v\right\|$, which holds by Lemma 6.2.3, thus the claim is settled. Therefore, in order to find a lower bound for θ, we focus on the case $\|v'\| = \|v\|$: for shortness, we let $\hat{\theta} = \mathrm{ang}(v', v)$. We have

$$\cos(\theta) = \frac{\langle v'-v, v\rangle}{\|v'-v\|\|v\|} = \frac{\langle v', v\rangle - \|v\|^2}{\|v'-v\|\|v\|} = \frac{\|v\|^2(\cos(\hat{\theta})-1)}{\|v'-v\|\|v\|} = \frac{\|v\|(\cos(\hat{\theta})-1)}{\|v'-v\|}.$$

Since

$$\begin{aligned}
\|v'-v\| = & \sqrt{\|v'-v\|^2} = \sqrt{\|v'\|^2 + \|v\|^2 - 2\|v'\|\|v\|\cos(\hat{\theta})} = \\
= & \sqrt{2\|v\|^2(1-\cos(\hat{\theta}))} = \|v\|\sqrt{2(1-\cos(\hat{\theta}))},
\end{aligned}$$

and because $\cos(\hat{\theta}) \geq 0$, we obtain

$$\cos(\theta) = \frac{\|v\|(\cos(\hat{\theta})-1)}{\|v\|\sqrt{2(1-\cos(\hat{\theta}))}} = \frac{-\sqrt{(1-\cos(\hat{\theta}))}}{\sqrt{2}} \geq -\frac{1}{\sqrt{2}}.$$

∎

We are now going to show that, for any word $\omega \in \Gamma^*$, the evolution $V(\omega)$ only increases distances between vectors by a constant factor, which does not depend on the length of the word ω.

Lemma 6.2.5. *For every $v, v' \in \mathcal{B}_1$ and $\omega = \sigma_1 \cdots \sigma_n \in \Gamma^*$, by letting*

$$\varphi = v \otimes v \otimes \alpha \quad \text{and} \quad \varphi' = v' \otimes v' \otimes \alpha,$$

it holds

$$\|\varphi' V(\omega) - \varphi V(\omega)\| \leq 4\|\varphi' - \varphi\|.$$

Proof. Without loss of generality, we can assume that $\|v'\| \geq \|v\|$. Moreover, we can assume that the angle $\mathrm{ang}(v, v')$ between the two vectors v and v' is not greater than $\frac{\pi}{2}$, because if it is, we can consider the vector $-v'$ instead of v', for which it holds $\mathrm{ang}(v, -v') \leq \frac{\pi}{2}$, and the proof is still valid, since $(-v') \otimes (-v') \otimes \alpha = v' \otimes v' \otimes \alpha = \varphi'$.

By letting $\Delta = v' - v$, we have

$$\begin{aligned}
\varphi' - \varphi =& (v + \Delta) \otimes (v + \Delta) \otimes \alpha - v \otimes v \otimes \alpha \\
=& v \otimes v \otimes \alpha + v \otimes \Delta \otimes \alpha + \Delta \otimes v \otimes \alpha + \Delta \otimes \Delta \otimes \alpha - v \otimes v \otimes \alpha \\
=& v \otimes \Delta \otimes \alpha + \Delta \otimes v \otimes \alpha + \Delta \otimes \Delta \otimes \alpha,
\end{aligned}$$

we can rewrite the left side of the Lemma's inequality as

$$\begin{aligned}
\|\varphi' V(\omega) - \varphi V(\omega)\| =& \|(\varphi' - \varphi)V(\omega)\| \\
=& \|(v \otimes \Delta \otimes \alpha)V(\omega) + (\Delta \otimes v \otimes \alpha)V(\omega) + \\
&+ (\Delta \otimes \Delta \otimes \alpha)V(\omega)\| \\
\leq& \|(v \otimes \Delta \otimes \alpha)V(\omega)\| + \|(\Delta \otimes v \otimes \alpha)V(\omega)\| + \\
&+ \|(\Delta \otimes \Delta \otimes \alpha)V(\omega)\|. \qquad (6.6)
\end{aligned}$$

In order to simplify the above equation, we study the following generic form of $\|(v_1 \otimes v_2 \otimes \alpha)V(\omega)\|$, which can be written as

$$\left\| \sum_{y = y_1 \cdots y_n} v_1 \prod_{j=1}^{n} U(\sigma_j) P(y_j) \otimes v_2 \prod_{j=1}^{n} U(\sigma_j) P(y_j) \otimes \alpha M(y) \right\|,$$

where each y is a word in C^n, i.e., each y_j represents a possible result of the observation after evolving on the symbol σ_j. We can have an upper bound of the

above value by moving the sum out of the norm. Moreover, since D is a DFA, it holds $\|\alpha M(y)\| = 1$ for every $y \in C^*$, so we can write

$$\|(v_1 \otimes v_2 \otimes \alpha)V(\omega)\| \leq \sum_{y=y_1\cdots y_n} \left\| v_1 \prod_{j=1}^{n} U(\sigma_j)P(y_j) \right\| \cdot \left\| v_2 \prod_{j=1}^{n} U(\sigma_j)P(y_j) \right\|.$$

The right side of the above inequality can be seen as the inner product between two vectors $\tilde{v}_1, \tilde{v}_2$ of dimension $|C|^n$, where the y-th component of $\tilde{v}_1$ (resp. $\tilde{v}_2$) is $\left\| v_1 \prod_{j=1}^{n} U(\sigma_j)P(y_j) \right\|$ (resp. $\left\| v_2 \prod_{j=1}^{n} U(\sigma_j)P(y_j) \right\|$). By Cauchy-Schwarz inequality we have that $\langle \tilde{v}_1, \tilde{v}_2 \rangle \leq \|\tilde{v}_1\|\|\tilde{v}_2\|$. Therefore, by the definition of norm $\|v\| = \sqrt{\sum_y |(v)_y|^2}$, we obtain

$$\begin{aligned}
\|(v_1 \otimes v_2 \otimes \alpha)V(\omega)\| \leq & \sqrt{\sum_{y=y_1\cdots y_n} \|v_1 \prod_{j=1}^{n} U(\sigma_j)P(y_j)\|^2} \cdot \sqrt{\sum_{y=y_1\cdots y_n} \|v_2 \prod_{j=1}^{n} U(\sigma_j)P(y_j)\|^2} \\
= & \sqrt{\|v_1\|^2\|v_2\|^2} && \text{(by Lemma 6.2.2)} \\
= & \|v_1\|\|v_2\|.
\end{aligned}$$

By replacing v_1 and v_2 with the vectors involved in 6.6, we obtain

$$\|\varphi' V(\omega) - \varphi V(\omega)\| \leq 2\|v\|\|\Delta\| + \|\Delta\|^2.$$

We now analyze the right part of the Lemma's equation: we first observe that

$$\begin{aligned}
\|\varphi' - \varphi\|^2 = & \|v \otimes \Delta \otimes \alpha + \Delta \otimes v \otimes \alpha + \Delta \otimes \Delta \otimes \alpha\|^2 \\
= & \|v \otimes \Delta + \Delta \otimes v + \Delta \otimes \Delta\|^2 \\
= & \|v \otimes \Delta\|^2 + \|\Delta \otimes v\|^2 + \|\Delta \otimes \Delta\|^2 + \\
& + 2\langle v \otimes \Delta, \Delta \otimes v\rangle + 2\langle v \otimes \Delta, \Delta \otimes \Delta\rangle + 2\langle \Delta \otimes v, \Delta \otimes \Delta\rangle \\
= & \|v\|^2\|\Delta\|^2 + \|\Delta\|^2\|v\|^2 + \|\Delta\|^2\|\Delta\|^2 + \\
& + 2\langle v, \Delta\rangle\langle \Delta, v\rangle + 2\langle v, \Delta\rangle\langle \Delta, \Delta\rangle + 2\langle \Delta, \Delta\rangle\langle v, \Delta\rangle \\
= & 2\|v\|^2\|\Delta\|^2 + \|\Delta\|^4 + 2(\langle v, \Delta\rangle)^2 + 4\|\Delta\|^2\langle v, \Delta\rangle
\end{aligned}$$

By letting $\theta = \text{ang}(v, \Delta)$, we can write $\langle v, \Delta\rangle = \|v\|\|\Delta\|\cos\theta$:

$$\|\varphi' - \varphi\|^2 = 2\|v\|^2\|\Delta\|^2 + \|\Delta\|^4 + 2\|v\|^2\|\Delta\|^2(\cos(\theta))^2 + 4\|v\|\|\Delta\|^3\cos(\theta).$$

Therefore, in order to prove the desired property, it is enough to show that

$$\begin{aligned}
(2\|v\|\|\Delta\| + \|\Delta\|^2)^2 \leq 16\ & (2\|v\|^2\|\Delta\|^2 + \|\Delta\|^4 + \\
& + 2\|v\|^2\|\Delta\|^2(\cos(\theta))^2 + 4\|v\|\|\Delta\|^3\cos(\theta)),
\end{aligned}$$

which is equivalent to

$$0 \leq 32\|v\|^2\|\Delta\|^2 + 16\|\Delta\|^4 + 32\|v\|^2\|\Delta\|^2(\cos(\theta))^2 + \\ + 64\|v\|\|\Delta\|^3\cos(\theta) - 4\|v\|^2\|\Delta\|^2 - \|\Delta\|^4 - 4\|v\|\|\Delta\|^3.$$

We can divide by $\|\Delta\|^2$ because if $\|\Delta\| = 0$ the inequality is trivially verified

$$15\|\Delta\|^2 + 4\|v\|(16\cos(\theta) - 1)\|\Delta\| + 4\|v\|^2(8\,(\cos(\theta))^2 + 7) \geq 0.$$

When solving for $\|\Delta\|$, the above inequality is always true if it holds

$$4\|v\|^2(16\cos(\theta) - 1)^2 - 60\|v\|^2(8\ (\cos(\theta))^2 + 7) \leq 0.$$

If $\|v\| = 0$, the inequality is clearly verified, otherwise we can write

$$17(\cos(\theta))^2 - 4\cos(\theta) - 13 \leq 0. \tag{6.7}$$

The left side of the above inequality is a quadratic function for $-1 \leq \cos(\theta) < 1$. Moreover, we recall that, at the beginning of this proof, we assumed that $\|v'\| \geq \|v\|$ and $\text{ang}(v, v') \leq \frac{\pi}{2}$, which allows us to use Lemma 6.2.4 to state that the maximum value of $17(\cos(\theta))^2 - 4\cos(\theta) - 13$ is for $|\theta| = \frac{3}{4}\pi$, so the left side of Equation 6.7 is bounded above by $4\sqrt{2} - 9$ which is a negative value, therefore the property holds. ∎

Conversion to DFA:

We now construct a deterministic automaton D_Ω equivalent to Ω, starting from the real-linear representation $\langle \varphi_0, \{V(\sigma)\}_{\sigma\in\Gamma}, \eta\rangle$. For any word $\omega \in \Gamma^*$, let φ_ω be the vector reached by the QFC Ω after reading ω, i.e., $\varphi_\omega = \varphi_0 V(\omega)$. We define a relation $\sim$ on the set of states reachable by Ω $\{\varphi_\omega \mid \omega \in \Gamma^*\}$ such that

$$\varphi_\omega \sim \varphi_{\omega'}$$
$$\Updownarrow$$
$$\exists\ \omega = \omega_1, \omega_2, \ldots, \omega_n = \omega' \in \Gamma^* \text{ such that } \|\varphi_{\omega_i} - \varphi_{\omega_{i+1}}\| < \frac{\delta}{4q\sqrt{k}}.$$

We point out that $\sim$ is an equivalence relation:

- $\varphi_\omega \sim \varphi_\omega$ trivially holds for any $\omega \in \Gamma^*$,
- if $\varphi_\omega \sim \varphi_{\omega'}$ and $\{\omega_k\}_{k=1,\ldots,n}$ is the sequence of words witnessing the relation, then the sequence witnessing $\varphi_{\omega'} \sim \varphi_\omega$ is $\{\omega_{n-k}\}_{k=1,\ldots,n}$,

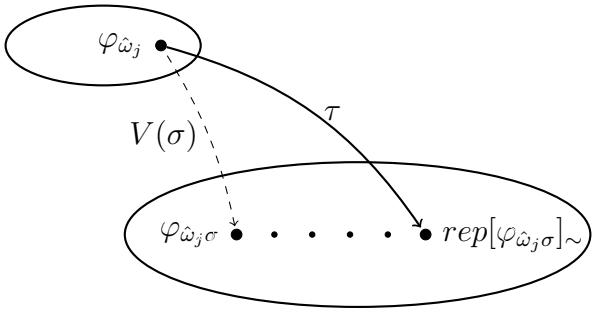


Figure 6.2: Evolution τ over a symbol σ. The dots represent quantum state vectors, while the ellipses indicate equivalence classes of $\sim$. The smaller points between $\varphi_{\hat{\omega}_j\sigma}$ and $rep[\varphi_{\hat{\omega}_j\sigma}]_\sim$ represent the vectors at distance smaller than $\frac{\delta}{4q\sqrt{k}}$ that witness the relation $\sim$ between them. The dashed arrow indicates the original quantum evolution, while the full arrow represents the behavior of the DFA.

- if $\varphi_\omega \sim \varphi_{\omega'}$ and $\varphi_{\omega'} \sim \varphi_{\omega''}$, by calling $\{\omega_k\}_{k=1,\dots,n}$ and $\{\omega_k\}_{k=n,\dots,n+m}$ the sequences of words witnessing the two relations, respectively, then the sequence witnessing $\varphi_\omega \sim \varphi_{\omega''}$ is $\{\omega_k\}_{k=1,\dots,n+m}$.

As we will see in the proof of Theorem 6.2.7 when discussing the descriptional complexity of D_ω, we have that the relation $\sim$ is of finite index. We choose a representative $rep[\varphi_\omega]_\sim$ for each equivalence class, we call those representatives

$$\varphi_{\hat{\omega}_1}, \varphi_{\hat{\omega}_2}, \dots, \varphi_{\hat{\omega}_s},$$

and we construct our DFA D_Ω as follows:

- the set of states coincides with the set of representative quantum state vectors $\{\varphi_{\hat{\omega}_1}, \varphi_{\hat{\omega}_2}, \dots, \varphi_{\hat{\omega}_s}\}$,
- the initial state is the vector $\varphi_{\hat{\omega}_1} = rep[\varphi_\epsilon]_\sim$,
- the evolution is $\tau(\varphi_{\hat{\omega}_j}, \sigma) = rep[\varphi_{\hat{\omega}_j\sigma}]_\sim$, intuitively shown in Figure 6.2,
- the final states are the representative vectors $\{\varphi_{\hat{\omega}_j} \mid \varphi_{\hat{\omega}_j}\eta > \lambda\}$ associated with words which were accepted in the original QFC.

Theorem 6.2.6. *D_Ω recognizes the same language as Ω.*

Proof. Let

- $z = z_1 z_2 \cdots z_n$ be the input word, $z_{\{j\}} = z_1 z_2 \cdots z_j$ the prefix of z of length j, and $z_{\{-j\}} = z_{j+1} z_{i+2} \cdots z_n$ the suffix obtained by removing $z_{\{j\}}$ from z,

- $\rho_j = \tau(\varphi_{w_1}, z_{\{j\}})$ be the vector reached by D_Ω after reading the first j symbols, therefore $\rho_0 = \varphi_{\omega_1}$ is the initial state of D_Ω,
- $\psi_j = \rho_{j-1}V(z_j)$ be the vector reached with $j-1$ steps of D_Ω followed by one step of Ω, therefore $\psi_0 = \varphi_0$ is the initial state of Ω.

Note that, for each $0 \leq j \leq n$, it holds $\psi_j \sim \rho_j$, because $\rho_j = rep[\psi_j]_\sim$. Moreover, by definition, the vectors witnessing $\psi_j \sim \rho_j$ are reachable in Ω. More formally, there exist $\psi_j = \gamma_{j,1}, \gamma_{j,2}, \ldots \gamma_{j,\ell_j} = \rho_j$ such that $\|\gamma_{j,i} - \gamma_{j,i+1}\| < \frac{\delta}{4q\sqrt{k}}$, and there exist $x_{j,t} \in \Gamma^*$ such that $\varphi_0 V(x_{j,t}) = \gamma_{j,t}$.

We first observe that, for all $0 \leq j \leq n$ and for all $1 \leq t \leq \ell_j$, it holds

$$\|\gamma_{j,t}V(z_{\{-j\}}) - \gamma_{j,(t+1)}V(z_{\{-j\}})\| < \frac{\delta}{q\sqrt{k}}, \tag{6.8}$$

as a consequence of Lemma 6.2.5. Moreover, since it holds

$$\rho_j V(z_{\{-j\}}) = \psi_{j+1}V(z_{\{-(j+1)\}}),$$

for all j's, Equation 6.8 implies that the vectors $\rho_j V(z_{\{-j\}})$ form a chain of vectors from the final configuration φ_z of Ω to the final configuration ρ_n of D_Ω, where the distance between each pair of consecutive vectors is smaller than the isolation of Ω. For an intuitive vision of this chain, see Figure 6.3.

We first show that

$$z \in L_\Omega \;\Rightarrow\; \tau(\varphi_{\omega_0}, z) \in F,$$

which is equivalent to showing

$$\varphi_0 V(z)\eta \geq \lambda + \delta \;\Rightarrow\; \rho_n \eta \geq \lambda + \delta. \tag{6.9}$$

Note that $\varphi_0 = \gamma_{0,1}$ and $\rho_n = \gamma_{n,\ell_n}$ and all $\gamma_{j,t}$'s are reachable in Ω through some word $x_{j,t}$, i.e., $\gamma_{j,t}V(z_{\{-j\}}) = \varphi_0 V(x_{j,t} \cdot z_{\{-j\}})$.

Since λ is a δ-isolated cut point, it holds

$$\gamma_{j,t}V(z_{\{-j\}})\eta \begin{cases} \geq \lambda + \delta & \text{if } x_{j,t}z_{\{-j\}} \in L_\Omega, \\ \leq \lambda - \delta & \text{if } x_{j,t}z_{\{-j\}} \notin L_\Omega. \end{cases}$$

If (6.9) did not hold, there would be a position in the bottom chain of Figure 6.3 where the acceptance probability associated to a vector in the chain is higher than the cut point, while the acceptance probability associated to its right neighbor is lower than the cut point. More formally, there would exist ι, κ such that

$$\gamma_{\iota,\kappa}V(z_{\{-\iota\}})\eta \geq \lambda + \delta \qquad \text{and} \qquad \gamma_{\iota,(\kappa+1)}V(z_{\{-\iota\}})\eta \leq \lambda - \delta,$$

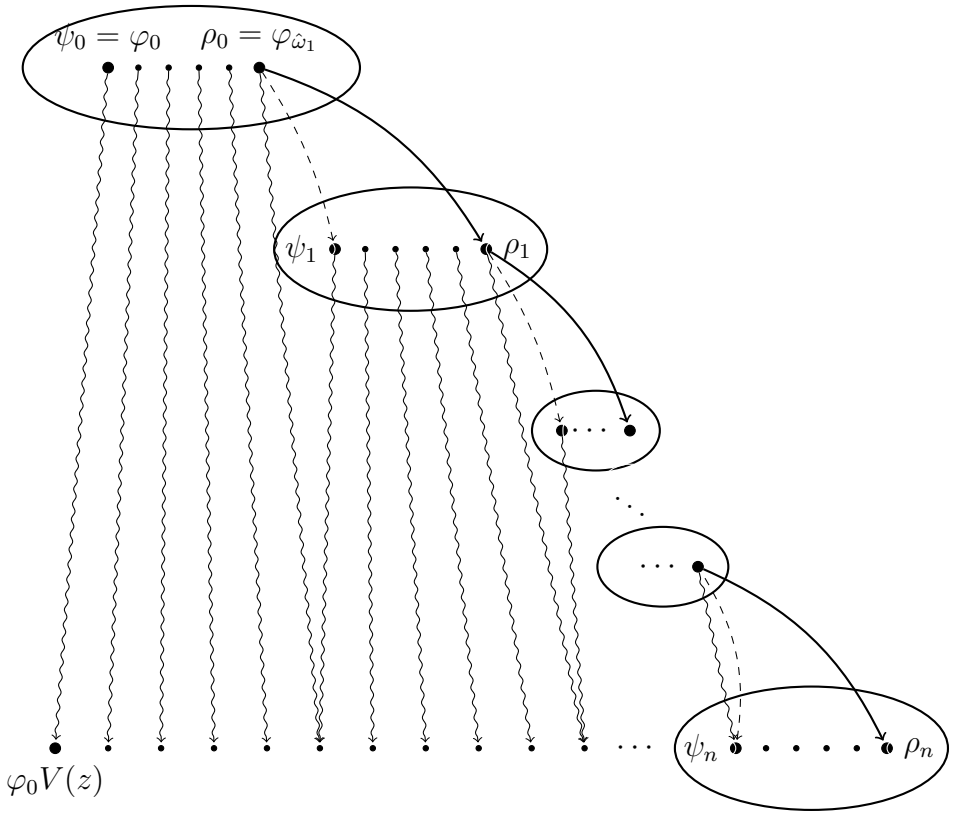


Figure 6.3: Evolution scheme of the computation over the word z. The full arrows describe the transitions of the DFA, while the snake arrows denote the quantum evolution from each vector $\gamma_{j,t}$ in the equivalence class reached after j symbols, through the dynamic V over the remaining suffix $z_{\{-j\}}$, leading to the vector $\gamma_{j,t}V(z_{\{-j\}})$ to a vector in the bottom chain. In this line, the leftmost point denotes the vector reached by Ω after reading z, while the rightmost point is the state reached by D_Ω after reading z. The intuitive idea for showing the correctness of D_Ω is that all the vectors in the bottom chain are close enough to their neighbor to be either all accepting or all rejecting states in the original QFC.

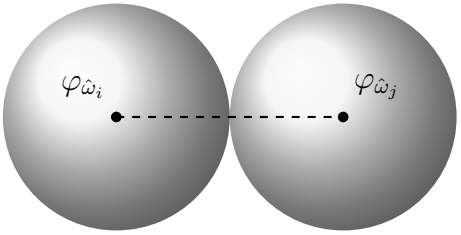


Figure 6.4: The minimal distance between the representative vectors of two different classes of the equivalence $\sim$ is $\frac{\delta}{4q\sqrt{k}}$. This implies that, by considering the set of balls $\mathcal{B}_{\frac{\delta}{8q\sqrt{k}}}(\varphi_{\hat{\omega}_i})$, for all $1 \leq i \leq s$, those balls are pairwise disjoint.

from which we would have

$$\begin{aligned}
2\delta \leq & \|(\gamma_{\iota,\kappa} V(z_{\{-\iota\}}) - \gamma_{\iota,(\kappa+1)} V(z_{\{-\iota\}}))\eta\| \\
\leq & \|\gamma_{\iota,\kappa} V(z_{\{-\iota\}}) - \gamma_{\iota,(\kappa+1)} V(z_{\{-\iota\}})\| \|\eta\| \\
\leq & \|\gamma_{\iota,\kappa} V(z_{\{-\iota\}}) - \gamma_{\iota,(\kappa+1)} V(z_{\{-\iota\}})\| \cdot \sqrt{4q^2 k} \\
< & \frac{\delta}{q\sqrt{k}} \cdot 2q\sqrt{k} = 2\delta
\end{aligned}$$

which generates a contradiction.

Symmetrically, one can show that $z \notin L_\Omega \Rightarrow \tau(\varphi_{\omega_0}, z) \notin F$, so the proof of correctness is complete. ∎

Descriptional complexity:

We now analyze the cost of the previously described conversion from QFC to DFA in terms of the number of states:

Theorem 6.2.7. *For any* QFC Ω *with q quantum states and k classical states, and a δ-isolated cut point, there exists an equivalent* DFA D_Ω *with m states, such that*

$$qk \geq \sqrt[3]{\frac{\log(m)}{4\log\left(\frac{9}{\delta}\right)}}.$$

Proof. Let $\langle \varphi_0, \{V(\sigma)\}_{\sigma\in\Gamma}, \eta \rangle$ be the real-linear representation of Ω. Clearly, the vector φ_0 can be seen as an element of $\mathbb{R}^d$, where $d = 2q \cdot 2q \cdot k$. When constructing the equivalent DFA D_Ω as described above, the number of states of D_Ω is exactly the number of equivalence classes $[\ ]_\sim$. We consider the sphere of radius $\frac{\delta}{8q\sqrt{k}}$ centered in the representative $\varphi_{\hat{\omega}_i}$. It is clear that such a sphere is disjoint from the analogous sphere centered in $\varphi_{\hat{\omega}_j}$, for $j \neq i$ (see Figure 6.4). Moreover, such spheres are all contained in the ball $\mathcal{B}_{1+\frac{\delta}{8q\sqrt{k}}}(0)$ of radius $1 + \frac{\delta}{8q\sqrt{k}}$ centered in 0, and the number of such spheres is exactly the number of equivalence classes of

$\sim$. The fact that any ball of fixed radius in $\mathbb{R}^d$ is a compact set implies that $\sim$ is of finite index, because if there existed an infinite number of disjoint spheres of radius $\frac{\delta}{8q\sqrt{k}}$ in $\mathcal{B}_{1+\frac{\delta}{8q\sqrt{k}}}(0)$, then it would not be totally bounded. Since the volume of a d-dimensional sphere of radius r is Kr^d, for a suitable constant K, which depends on d, there exist at most

$$\frac{K(1+\delta/8q\sqrt{k})^d}{K(\delta/8q\sqrt{k})^d} = \left(1+\frac{8q\sqrt{k}}{\delta}\right)^{4q^2k}$$

spheres, and this number is an upper bound for the number m of states of D_Ω. Since $\delta \leq 1$, for any $q, k \geq 1$ we have

$$\begin{aligned}\log(m) &\leq 4q^2k\left(\log\left(1+\frac{8q\sqrt{k}}{\delta}\right)\right)\\ &\leq 4q^2k\left(\log\left(\frac{9q\sqrt{k}}{\delta}\right)\right)\\ &\leq 4q^2k\left(\log\left(\frac{9}{\delta}\right)+\log(q\sqrt{k})\right).\end{aligned}$$

We notice that

$$\log\left(\frac{9}{\delta}\right)+\log(q\sqrt{k}) \leq q\sqrt{k}\log\left(\frac{9}{\delta}\right),$$

because the function $\log(9/\delta)(1-x)+\log(x)$ is non positive for any $\delta \leq 1$ and $x \geq 1$, so we can write

$$\log(m) \leq 4q^2k\left(q\sqrt{k}\log\left(\frac{9}{\delta}\right)\right) \leq 4(qk)^3\log\left(\frac{9}{\delta}\right).$$

∎

Since any MM-QFA can be simulated by a QFC without increasing the number of quantum states and using only 3 classical states, the above result implies the following

Corollary 6.2.8. *For any* MM-QFA Ω *of size* q*, and a* δ*-isolated cut point, there exists an equivalent* DFA D_Ω *with* m *states, such that*

$$q \geq \sqrt[3]{\frac{\log(m)}{108\log\left(\frac{9}{\delta}\right)}}.$$

As a final remark, we notice that the technique used for proving Theorem 6.2.7 is an adaptation of Rabin's technique used in [49] and [12] for obtaining similar conversion bounds on PFAs and MO-QFAs, respectively. However, both in

the case of PFAs and MO-QFAs, the equivalence relation determining the states of the DFA was in fact a congruence, so the proof for the correctness of the DFA was straightforward. In the case of QFC, instead, the relation $\sim$ is not a congruence, so we had to ensure that, starting from two different quantum states in the same class, after the evolution on the same word, the two resulting states still have the property of being either both accepting or both rejecting, even if they belong to different classes. This was possible because of the property proved in Lemma 6.2.5 and by choosing an appropriate distance value for the definition of $\sim$. In [37], the authors state, for MM-QFAs, a property similar to Lemma 6.2.5 on an operator T that represents both the evolution and the measurement performed after reading the input symbol. They state the existence of a constant c that bounds the increase in the distance between state vectors after the application of the operator T. However, the lack of linearity of the operator T makes it hard to calculate a numerical value of c. The decision of approaching the problem in the more general case of QFCs allowed us use a linearity argument.

Chapter 7

Periodicity problems concerning the behavior of quantum automata

This Chapter is devoted to the behavior of unary MM-QFAs; the results we present here were published in [16, 17]. The first problem we shall be dealing with concerns the structure of the phase space of MM-QFAs. In [3], the existence of an *ergodic subspace* and a *transient subspace* for MM-QFAs is emphasized. These subspaces somehow represent the quantum counterparts of ergodic and transient states in Markov chains. However, in the quantum case, it is not possible to single out these two subspaces by using graph theoretical tools: while for DFAs, NFAs and PFAs we often used graphs to describe the system, in the quantum case the graph associated to the system evolution is not representative of the state reachability relation. In fact, due to the presence of observations and complex amplitudes, depending on how the system is measured, the probability of reaching a state q_k from a state q_j might be zero even if there is a path from q_j to q_k (see Figure 7.1). This interference phenomenon is the reason why we do not use graph theory when treating ergodicity. In fact, the decision problems studied in this Chapter, which would be trivially solved through graph theory in the classical case, require a completely different approach. Here, we provide an algorithm to determine the dimension of the ergodic and transient subspaces by computing the degree of particular polynomials. The algorithm works in polynomial time whenever it runs on MM-QFAs whose amplitudes are complex numbers with rational components.

Then, we focus on periodicity by investigating the following decision problem: given a unary MM-QFA A and a positive integer d, is p_A a function of period d? We show that this problem is decidable by exhibiting an algorithm that works in polynomial time if the amplitudes of transition matrices and initial superposition of A are complex numbers with rational components. We remark that MM-QFAs with rational entries are able to exhibit sophisticated behaviours (see, e.g.,

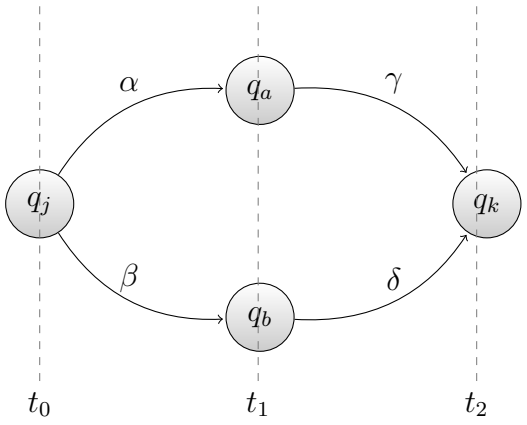


Figure 7.1: Interference phenomenon in a quantum automaton. Suppose that at time t_0 the system is in the basis state q_j. If we perform a measurement in both the following steps, at time t_1 the automaton will collapse to state q_a with probability $|\alpha|^2$ or to state q_b with probability $|\beta|^2$, while at time t_2 the probability of being observed in state q_k is $|\alpha|^2|\gamma|^2+|\beta|^2|\delta|^2$. If, however, we leave the evolution untouched at time t_1 without performing any measurement and we only observe at time t_2, the probability that the system collapses to state q_k is $|\alpha\gamma+\beta\delta|^2$, which can be zero even if $\alpha, \beta, \gamma, \delta$ are nonzero values.

Section 7.2). Thus, restricting problems to this kind of devices is well worth investigating.

7.1 Nonhalting space of MM-QFAs

In order to have a better understanding of the form of the events induced by unary quantum automata, we focus on an important feature related to the dynamics of MM-QFAs. Given a unary MM-QFA

$$\Omega = \langle \varphi_0,\ U(\sigma),\ \mathcal{O} = aP(a) + rP(r) + gP(g) \rangle,$$

we call $E_h = \mathbb{C} \cdot (P(a) + P(r))$ the *halting space* and $E_g = \mathbb{C} \cdot P(g)$ the *nonhalting space* of Ω. As a key step in proving that certain languages cannot be accepted by MM-QFAs, Ambainis and Freivalds in [3] show that the non halting space E_g of a MM-QFA on a general alphabet can be decomposed into two orthogonal subspaces having certain properties. In the unary case, this result can be stated as follows:

Lemma 7.1.1. [3] *There exist two subspaces E_1, E_2 of E_g such that $E_g = E_1 \oplus E_2$, and, for any unitary complex vector φ,*

(i) if $\varphi \in E_1$, then $\varphi U(\sigma)P(g) \in E_1$ and $\|\varphi U(\sigma)P(g)\| = \|\varphi\|$;

(ii) if $\varphi \in E_2$, then $\|\varphi(U(\sigma)P(g))^k\| \to 0$, for $k \to \infty$.

By using a terminology similar to that used in the classification of the states of Markov chains, we call E_1 *ergodic space* and E_2 *transient space*. However, in the quantum realm, it is not possible to use graph theoretic arguments to single out these two subspaces. As an example, consider a unary MM-QFA having transition matrix $U(\sigma)$ and non halting projector $P(g)$ defined as follows:

$$U(\sigma) = \begin{bmatrix} \frac{\sqrt{2}+1}{2\sqrt{2}} & \frac{\sqrt{2}-1}{2\sqrt{2}} & \frac{1}{2} \\ \frac{\sqrt{2}-1}{2\sqrt{2}} & \frac{\sqrt{2}+1}{2\sqrt{2}} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} \end{bmatrix}, \qquad P(g) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The restriction of $U(\sigma)P(g)$ to E_g is the matrix U_g, obtained from $U(\sigma)P(g)$ by deleting the third row and column, i.e., the rows and columns corresponding to the halting states. Similarly to Markov chains (see, e.g., [47]), we can represent transitions among go states by a 2-vertex digraph whose weight matrix is exactly U_g. Although such a digraph clearly consists of a single strongly connected component, we have that E_g is partitioned into the ergodic space E_1 and the orthogonal transient space E_2, both of dimension 1. In fact, it is easy to see that the eigenvalues of U_g are $\nu_1 = 1$ and $\nu_2 = 1/\sqrt{2}$ and the corresponding eigenspaces are $E_1 = \mathbb{C}(1/\sqrt{2}, 1/\sqrt{2})$ and $E_2 = \mathbb{C}(-1/\sqrt{2}, 1/\sqrt{2})$. Easy calculations show that E_1 and E_2 satisfy Lemma 7.1.1.

We are now going to provide an algorithm for determining the dimension of E_1 and E_2 for a generic unary MM-QFA. We denote by $q_M(\nu) = \det(M - \nu I)$ the characteristic polynomial of a square matrix M. The key idea of our algorithm comes from the properties of the subspaces E_1 and E_2 given in Lemma 7.1.1. Let U_g be the restriction of $U(\sigma)P(g)$ to the non halting space E_g. As stated in Lemma 7.1.1(i), E_1 is U_g-*invariant*, i.e., vectors in E_1 are mapped by U_g into vectors in E_1. Moreover, it is not hard to see that Lemma 7.1.1(i) and (ii) implies that E_2 is U_g-invariant as well. It is a well know result (see, e.g., [52]) that, given a space $V = V_1 \oplus V_2$ and a matrix T on V, if V_1 and V_2 are T-invariant then there exists an invertible matrix S such that

$$T = S \begin{pmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{pmatrix} S^{-1},$$

where:

- the dimension of A_1 (resp., A_2) is the dimension of V_1 (resp., V_2);
- the matrix A_1 (resp., A_2) is the transformation T restricted to V_1 (resp., V_2);
- the eigenvalues of T on V are given by the eigenvalues of A_1 plus those of A_2, more precisely, $q_T(\nu) = q_{A_1}(\nu)q_{A_2}(\nu)$.

By this result, we let

$$U_g = S \begin{pmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{pmatrix} S^{-1},$$

for suitable matrices S, A_1, A_2 with A_1 (resp., A_2) acting on E_1 (resp., E_2). By Lemma 7.1.1, we have that the modulus 1 eigenvalues of U_g are exactly the eigenvalues of A_1 (by also taking into account eigenvalues multiplicity). So, our algorithm computes the dimension of A_1, which is the dimension of E_1, by counting the modulus 1 eigenvalues of U_g. The running time of the algorithm turns out to be polynomial whenever it works on MM-QFAs with *rational entries*, i.e., complex numbers with rational components.

Now we describe how our algorithm works on the unary MM-QFA $A = \langle \varphi_0, \{U(\sigma), U(\#)\}, \mathcal{O} \rangle$ as input. As a first step, according to Theorem 5.4.14, we transform A into an equivalent MM-QFA $A' = \langle \varphi_0', \{M(\sigma), M(\#)\}, \mathcal{O}' \rangle$ described by real entries. Let U_g (resp., M_g) be the matrix obtained from $U(\sigma)$ (resp., $M(\sigma)$) by deleting all the rows and columns related to the halting states. Clearly, if $U_g \in \mathbb{C}^{\mu \times \mu}$ then $M_g \in \mathbb{R}^{2\mu \times 2\mu}$. The following holds:

Lemma 7.1.2. *If $\nu_1, \nu_2, \ldots, \nu_\mu$ are the eigenvalues of U_g, then the eigenvalues of M_g are $\nu_1, \nu_2, \ldots, \nu_\mu$, and $\nu_1^*, \nu_2^*, \ldots, \nu_\mu^*$.*

Proof. Let $U_g = A + i\,B$, for suitable matrices $A, B \in \mathbb{R}^{\mu \times \mu}$. By rearranging rows and columns, we can transform M_g into

$$\bar{M}_g = \begin{pmatrix} A & B \\ -B & A \end{pmatrix}.$$

More precisely, there exists a permutation matrix X such that $\bar{M}_g = X M_g X^T$, where X^T performs on columns the same permutation induced by X on rows. Clearly, X^T is the inverse of X, and this implies that $\bar{M}_g$ and M_g are similar (see Section 1.1.2). Now, consider the unitary matrix

$$Y = \frac{1}{\sqrt{2}} \begin{pmatrix} I & -iI \\ -iI & I \end{pmatrix},$$

and let

$$\hat{M}_g = Y \bar{M}_g Y^\dagger = \begin{pmatrix} A + iB & 0 \\ 0 & (A + iB)^* \end{pmatrix}.$$

This shows that $\hat{M}_g$ and $\bar{M}_g$ are similar as well. Since similar matrices have the same characteristic polynomial, we get $q_{M_g}(\nu) = q_{\bar{M}_g}(\nu) = q_{\hat{M}_g}(\nu)$. Easy calculations show that

$$q_{M_g}(\nu) = q_{U_g}(\nu) \cdot q_{U_g^*}(\nu) = q_{U_g}(\nu) \cdot (q_{U_g}(\nu))^*,$$

whence the result. ∎

So, we compute $q_{M_g}(\nu) = \sum_{k=0}^{2\mu} c_k \nu^k$, and define the polynomial

$$\tilde{q}_{M_g}(\nu) = \sum_{k=0}^{2\mu} c_k \nu^{2\mu-k}$$

having as roots the reciprocals of the nonzero roots of $q_{M_g}(\nu)$, i.e., if $\nu' \neq 0$ is a root of $q_{M_g}(\nu)$ then $\frac{1}{\nu'}$ is a root of $\tilde{q}_{M_g}(\nu)$ and viceversa. It is not hard to see that the nonzero eigenvalues of U_g are in the form $\nu_j = \rho\, \mathrm{e}^{i\theta_j}$, where $|\rho| \leq 1$. Then, by Lemma 7.1.2, the following properties hold:

- If $\nu_j = \mathrm{e}^{i\theta_j}$ then $\nu_j^* = \mathrm{e}^{-i\theta_j} = \frac{1}{\nu_j}$ is a root of $q_{M_g}(\nu)$. Therefore ν_j is a root of $\tilde{q}_{M_g}(\nu)$.
- If $|\nu_j| < 1$ then $\left|\frac{1}{\nu_j}\right| > 1$. Therefore $\frac{1}{\nu_j}$ is not a root of $q_{M_g}(\nu)$, and so ν_j is not a root of $\tilde{q}_{M_g}(\nu)$.

This implies that the roots of $q_{M_g}(\nu)$ of modulus 1 are exactly those ν_j's that are roots of both $q_{M_g}(\nu)$ and $\tilde{q}_{M_g}(\nu)$. To count them, we compute the greatest common divisor $h_{M_g}(\nu)$ of the polynomials $q_{M_g}(\nu)$ and $\tilde{q}_{M_g}(\nu)$. The degree τ of $h_{M_g}(\nu)$ is twice the number of the eigenvalues of U_g of modulus 1, and therefore it is twice the dimension of E_1. As a consequence, the dimension of E_2 is $\mu - \frac{\tau}{2}$. Concerning the time complexity of our algorithm, one may observe that traditional operations on matrices and polynomials are involved. Indeed, if A has rational entries, such operations are easily seen to be performed in polynomial time (see, e.g., [4]).

7.2 Establishing d-periodicity for MM-QFAs

In this section, we investigate the periodicity of the events induced by MO-QFAs and MM-QFAs. We notice that, in general, the presence of the ergodic component in the dynamic of a QFA does not necessarily imply the periodicity of the induced event. For instance, consider the unary MO-QFA

$$A = \langle (1,0), \begin{pmatrix} \cos\pi\theta & \sin\pi\theta \\ -\sin\pi\theta & \cos\pi\theta \end{pmatrix} = \begin{pmatrix} 3/5 & 4/5 \\ -4/5 & 3/5 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \rangle.$$

The event induced by A is $p_A(n) = (\cos \pi n\theta)^2$, which is periodic if and only if θ is rational. It is known (see, e.g., [46]) that, for rational θ, the only rational values of $\cos(\pi\theta)$ are $0, \pm\frac{1}{2}, \pm 1$. Therefore, p_A is clearly not periodic.

Here, we consider the following decision problem:
d-PERIODICITY

- Input: a unary MM-QFA A and an integer $d > 0$.
- Question: is p_A a d-periodic event?

One could treat this problem on MO-QFAs with first order logic techniques similar to the ones used in [12, 20], since the periodicity condition may be expressed through a first order formula ϕ over the reals, which can be decided effectively by Tarski-Seidenberg elimination methods. However, this approach is based on the fact that the closed semigroup generated by a MO-QFA's dynamics is compact, and therefore it coincides with the zero set of a finite set P of polynomials, hence ϕ is defined in terms of P. In the case of MM-QFAs, the semigroup generated by the system dynamics does not have the compactness property, so we use an alternative approach. Here we present an algorithm for deciding d-PERIODICITY such that, if the MM-QFAs in input are described by rational entries, the time complexity turns out to be polynomial.

First of all, we provide a new representation for Measure-Many automata. Given a MM-QFA

$$A = \langle \varphi_0, \{U(\sigma), U(\#)\}, \mathcal{O}_{int}, \mathcal{O}_{fin} \rangle$$

with m basis states and

- $\mathcal{O}_{int} = aP_{int}(a) + rP_{int}(r) + gP_{int}(g)$,
- $\mathcal{O}_{fin} = aP_{fin}(a) + rP_{fin}(r)$,

we define its *linear representation* as the tuple

$$\langle \varphi, V, \eta_1, \eta_2 \rangle,$$

where

- $\varphi = \varphi_0 \otimes \varphi_0^*$,
- $V = (U(\sigma)P_{int}(g)) \otimes (U(\sigma)P_{int}(g))^*$,
- $\eta_1 = (U(\sigma) \otimes U(\sigma)^*) \sum_{j=1}^{m} (P_{int}(a))_j \otimes (P_{int}(a))_j$,
- $\eta_2 = (U(\#) \otimes U(\#)^*) \sum_{j=1}^{m} (P_{fin}(a))_j \otimes (P_{fin}(a))_j$.

We have that $p_A(n) = \sum_{i=0}^{n-1} \varphi V^i \eta_1 + \varphi V^n \eta_2$, for every $n \in \mathbb{N}$. In fact:

$$\begin{aligned}
\varphi V^i \eta_1 &= \sum_{j=1}^{m} \Big(\varphi_0 \left(U(\sigma) P_{int}(g)\right)^i U(\sigma)(P_{int}(a))_j \Big) \otimes \\
&\qquad \otimes \Big(\varphi_0^* (U(\sigma) P_{int}(g))^{*i} U(\sigma)^* (P_{int}(a))_j \Big) \\
&= \sum_{j=1}^{m} \Big(\varphi_0 \left(U(\sigma) P_{int}(g)\right)^i U(\sigma) P_{int}(a) \Big)_j \cdot \Big(\varphi_0^* (U(\sigma) P_{int}(g))^{*i} U(\sigma)^* P_{int}(a) \Big)_j \\
&= \left\| \varphi_0 \left(U(\sigma) P_{int}(g)\right)^i U(\sigma) P_{int}(a) \right\|^2 .
\end{aligned}$$

Similarly, $\varphi V^n \eta_2 = \left\| \varphi_0 \left(U(\sigma) P_{int}(g)\right)^n U(\#) P_{fin}(a) \right\|^2$.

Now, we recall some useful tools. We need the notion of generating function:

Definition For a function $f : \mathbb{N} \to \mathbb{C}$, its *generating function* is defined as $G_f(z) = \sum_{k=0}^{+\infty} f(k) z^k$, for all $z \in \mathbb{C}$ such that $|z| < 1$.

We also need the following well known property of square matrices:

Lemma 7.2.1. *Let V be a complex square matrix such that $\lim_{n\to\infty} V^n = \mathbf{0}$. Then, the matrix $(I - V)^{-1}$ exists, and we have $\sum_{k=0}^{\infty} V^k = (I - V)^{-1}$.*

Let now $A = \langle \varphi, V, \eta_1, \eta_2 \rangle$ be a MM-QFA in linear representation with rational entries. By letting $\eta_3 = \eta_2 - \eta_1$, we express the d-periodicity condition

$$\forall n \in \mathbb{N} \quad p_A(n) = p_A(n + d)$$

as

$$\forall n\in\mathbb{N}\quad \sum_{i=1}^{n-1}\varphi V^i\eta_1+\varphi V^n\eta_2=\sum_{i=1}^{n+d-1}\varphi V^i\eta_1+\varphi V^{n+d}\eta_2$$

$$\Updownarrow$$

$$\forall n\in\mathbb{N}\quad \sum_{i=0}^{n-1}\varphi V^i\eta_1-\varphi\eta_1+\varphi V^n\eta_2=\sum_{i=0}^{n+d-1}\varphi V^i\eta_1-\varphi\eta_1+\varphi V^{n+d}\eta_2$$

$$\Updownarrow$$

$$\forall n\in\mathbb{N}\quad \sum_{i=0}^{n}\varphi V^i\eta_1-\varphi V^n\eta_1+\varphi V^n\eta_2=\sum_{i=0}^{n+d}\varphi V^i\eta_1-\varphi V^{n+d}\eta_1+\varphi V^{n+d}\eta_2$$

$$\Updownarrow$$

$$\forall n\in\mathbb{N}\quad \sum_{i=0}^{n}\varphi V^i\eta_1+\varphi V^n\eta_3=\sum_{i=0}^{n+d}\varphi V^i\eta_1+\varphi V^{n+d}\eta_3$$

$$\Updownarrow$$

$$\forall n\in\mathbb{N}\quad \varphi\left(\sum_{i=0}^{n}V^i\right)\eta_1-\varphi\left(\sum_{i=0}^{n+d}V^i\right)\eta_1=\varphi V^{n+d}\eta_3-\varphi V^n\eta_3.$$

$$\Updownarrow$$

$$\forall n\in\mathbb{N}\quad \varphi\left(\sum_{i=0}^{n}V^i\right)\eta_1-\varphi\left(\sum_{i=0}^{d-1}V^i\right)\eta_1-\varphi\left(V^d\sum_{i=0}^{n}V^i\right)\eta_1=\varphi(V^d-I)V^n\eta_3.$$

$$\Updownarrow$$

$$\forall n\in\mathbb{N}\quad \varphi(I-V^d)\sum_{i=0}^{n}V^i\eta_1-\varphi\sum_{i=0}^{d-1}V^i\eta_1=\varphi(V^d-I)V^n\eta_3.$$

By taking the generating functions of each term of the above equation, we obtain

$$\varphi(I-V^d)\sum_{k=0}^{+\infty}\left(\sum_{i=0}^{k}V^i\right)z^k\eta_1-\varphi\sum_{i=0}^{d-1}V^i\eta_1\sum_{k=0}^{+\infty}z^k=\varphi(V^d-I)\sum_{k=0}^{+\infty}(Vz)^k\eta_3. \quad (7.1)$$

Notice that, by rearranging terms, the two sums in the first term of the above equation can be rewritten as

$$\sum_{k=0}^{+\infty}\left(\sum_{i=0}^{k}V^i\right)z^k=\sum_{j=0}^{+\infty}z^j\sum_{k=0}^{+\infty}V^kz^k.$$

Moreover, since $V=(U(\sigma)P_I(g))\otimes(U(\sigma)P_I(g))^*$ with $U(\sigma)$ unitary and $P_I(g)$ projection, we have $(Vz)^n\to\mathbf{0}$ for $z\in\mathbb{C}$ with $|z|<1$ and $n\to\infty$. Therefore, we can apply Lemma 7.2.1 to Equation (7.1) and obtain

$$\frac{1}{1-z}\varphi(I-V^d)(I-Vz)^{-1}\eta_1-\frac{1}{1-z}\varphi\sum_{i=0}^{d-1}V^i\eta_1=\varphi(V^d-I)(I-Vz)^{-1}\eta_3. \quad (7.2)$$

For a matrix $M \in \mathbb{C}^{m\times m}$, we let its *adjugate* matrix $\mathrm{adj}(M)$ as $\mathrm{adj}(M)_{ij} = (-1)^{i+j}\det(M_{[ij]})$, where $M_{[ij]}$ is the matrix obtained from M by deleting the ith row and jth column. Since $M^{-1} = \frac{(\mathrm{adj}(M))^T}{\det(M)}$, Equation (7.2) becomes

$$\varphi(I - V^d)A(z)\eta_1 - \det(I - Vz)\cdot\varphi\sum_{i=0}^{d-1}V^i\eta_1 = (1-z)\cdot\varphi(V^d - I)A(z)\eta_3,$$

where $A(z) = (\mathrm{adj}(I - Vz))^T$. Notice that both terms of the above equation are rational polynomials of degree at most s, where $s \times s$ is the dimension of V. By multiplying these two polynomials by the least common multiple of the denominators of their coefficients, we reduce the original problem to testing the equality of two integer vectors of dimension s. All the operations involved in this algorithm require polynomial time (see, e.g. [4]).

As a final observation, we notice that our algorithm works also on MM-QFAs having algebraic entries. However, in this case the polynomial time complexity is not guaranteed. Therefore we get

Theorem 7.2.2. *d-PERIODICITY is decidable; the decision time is polynomial whenever the input QFA has rational entries.*

Chapter 8

Conclusion

In the previous chapters we presented many descriptional complexity issues on several models of finite state automata, showing their strength and weakness by analyzing families of unary languages defined by periodicity conditions. First, we extended the nondeterministic Chrobak normal form to PFAs, showing that, for the probabilistic case, if we want to guarantee minimality in the ergodic part, it is not enough to randomly choose which periodicity to check, but we need to be able to assign different probability values to different remainder classes of the same periodicity. We then used this form to make a descriptional complexity comparison of the three classical models, presenting the first example of a nontrivial family of languages which maximize the size of PFAs, i.e., the number of states required is exactly the same as the one needed for deterministic automata. Moreover, when probabilism helps reducing the size of DFAs, this is not necessarily the case where PFAs reach the state lower bound, nor the case where equivalent minimal NFAs reach the state upper bound (i.e., the size of equivalent DFAs [41]). In other words, we have shown the existence of infinitely many languages satisfying the following:

$$\text{LOWER BOUND} < \text{size of PFAs} < \text{size of NFAs} < \text{UPPER BOUND}.$$

For what concerns quantum automata, there are several results in the literature about the descriptional compexity of recognizing periodic languages on the model of MO-QFAs [3, 10, 30, 41]. In this thesis, we considered variants of QFAs more complex and powerful like MM-QFAs which, in the unary case, recognize all regular (i.e., ultimately periodic) languages. In particular, we constructively proved that any unary regular language can be recognized by a MM-QFA of size linear with respect to the one of the minimal DFA. We also singled out families of unary regular languages for which the size of the accepting MM-QFAs can be exponentially decreased. We showed that this is close to the biggest possible gap between

DFAs and the quantum models, by investigating size lower bounds for quantum devices. In order to have a generic approach to the study of lower bounds on the size of quantum devices, we considered the model of QFCs, which are able to simulate several known variants of QFA, like MO-QFAs and MM-QFAs. By suitably adapting Rabin's technique, we gave a lower bound for the cost of the conversion from QFC to DFA on alphabets of arbitrary size. Finally, as a preliminary step for approaching the problem of the simulation of unary MM-QFAs by classical devices, we analyzed the inner structure of MM-QFAs, which turns out to be more complex than the one of MO-QFAs. In particular, we singled out an ergodic and a transient component in the non halting subspace. These subspaces somehow represent the quantum counterparts of ergodic and transient states in Markov chains. We gave an algorithm, for computing the dimension of both these components. We notice that, the presence of an ergodic component does not necessarily lead to a periodic behavior. Thus, we designed an algorithm testing whether the event induced by a MM-QFA is periodic. These two algorithms run in polynomial time whenever the MM-QFA given in input has complex amplitudes with rational components.

Below, we provide some considerations about the results obtained so far and some ideas on how to extend them.

8.1 Classical automata

The normal form discussed in Chapter 3 guarantees a simple and compact structure for minimal unary PFAs, where probabilism is restricted to a single step of computation and the acceptance probability of the input words can be easily obtained by looking at one single transition for each cycle. On the other hand, this form does not preserve the original cut point nor the isolation. In fact, consider a prime $p > 2$ and the language $L_p = \{a^h \mid \langle h \rangle_p \neq 0\}$. The minimal DFA for L_p is made of a single cycle of length p, with $p-1$ accepting states, and it is also a PFA in Chrobak normal form accepting L_p exactly. However, the equivalent PFA in cyclic normal form, obtained with the conversion given in the proof of Theorem 3.2.1, has cut point $1/(2(p-1))$ isolated by $1/(2(p-1))$ (see Figure 8.1). The normal form we proposed is meant to be an instrument for analyzing the succinctness bounds of the generic model of PFA with isolated cut point, while for preserving recognition with high probability, one could consider relaxations of this normal form, for example by allowing multiple accepting states in each cycle.

The second issue about this normal form is that the increment of states required for the initial path cannot be bounded in terms of the size of the original

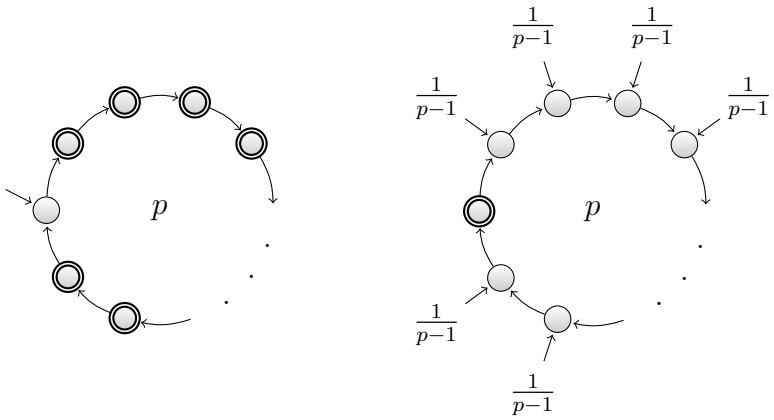


Figure 8.1: Two equivalent PFAs for L_p. The one on the left is also a DFA, so it accepts with certainty, while the one on the right is in cyclic normal form.

PFA. However, this limit is unavoidable for any normal form which aims to restrict probabilism in a single state. In fact the initial path corresponds to the transient part of the automaton, and the length of this part really depends on the probability associated to the loops in the transient part of the PFA, rather than on the number of states. It might be interesting to study whether it is possible to bound this length in terms of other parameters like, for example, the isolation of the cut point.

Another interesting problem is the extension of this normal form to the two-way probabilistic model, in analogy to what obtained for two-way NFAs [42]. Since two-way unary PFAs with isolated cut point still recognize only regular languages (as shown in [36]), having a normal form similar to the Chrobak or cyclic one could be helpful to obtain state bounds on the conversion between unary two-way PFAs and one-way DFAs.

8.2 Quantum automata

As we pointed out at the beginning of Chapter 7, in the quantum case the graph associated to the system evolution is not representative of the state reachability relation. Therefore the study of the descriptional complexity and the decision problems approached in Chapter 7 required techniques completely different from the classical case, where standard graph-theoretical tools are involved. The algorithm we presented for the problem d-PERIODICITY checks whether the event induced by a specific MM-QFA has a certain periodicity. It would be interest-

ing to find generic necessary and/or sufficient conditions on the QFA for deciding whether the induced event is periodic. For what concerns the study of the descriptional complexity, a natural question is to verify whether the lower bound stated in Theorem 6.2.7 is optimal. The bound obtained on QFCs carries over to more specific variants of quantum automata, such as MM-QFAs, so another interesting investigation could be analyzing for which variants of QFAs this bound can be improved. Finally, many problems on the computational power of quantum devices are still open: the set of languages recognized by MM-QFAs with isolated cut point working on alphabets of arbitrary size is still to be characterized, as well as the computational power of two-way quantum models, even in the unary case.

Bibliography

[1] A. Ambainis, A. Nayak, A. Ta-Shma, U. V. Vazirani. Dense quantum coding and quantum finite automata. *J. ACM* **49(4)**, pp. 496–511 (2002).

[2] A. Ambainis, M. Beaudry, M. Golovkins, A. Kikusts, M. Mercer, D. Thérien. Algebraic Results on Quantum Automata. *Theory of Computing Systems* **39(1)**, pp. 165–188 (2006).

[3] A. Ambainis, R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. *In Proc. 39th Annual Symposium on Foundations of Computer Science. IEEE Computer Society Press*, pp. 332–342 (1998).

[4] A.V. Aho, J.E. Hopcroft, J.D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, MA (1974).

[5] A. Ambainis, A. Kikusts, M. Valdats. On the Class of Languages Recognizable by 1-way Quantum Finite Automata. *In Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science. Springer*, pp. 305–316 (2001), Lecture Notes in Comput. Sci. 2010.

[6] P. Benioff. Quantum mechanical Hamiltonian models of Turing machines. *J. Stat. Phys.* **29**, pp. 515–546 (1982).

[7] A. Bertoni, M. Carpentieri. Regular languages accepted by quantum automata. *Information and Computation* **165**, pp. 174–182 (2001).

[8] A. Bertoni, C. Mereghetti. *Dispense del corso di informatica teorica.* Dispense per il corso di "Informatica teorica" per il CDL in Informatica, Università degli Studi di Milano.

[9] A. Bertoni, C. Mereghetti, B. Palano. Approximating stochastic events by quantum automata. *In pre-proceedings of ERATO Conference on Quantum Information Science* (2003).

[10] A. Bertoni, C. Mereghetti, B. Palano. Quantum Computing: 1-Way Quantum Automata. *Developments in Language Theory*, pp. 1–20 (2003).

[11] A. Bertoni, C. Mereghetti, B. Palano. Small size quantum automata recognizing some regular languages. *Theoretical Computer Science* **340**, pp. 394–407 (2005).

[12] A. Bertoni, C. Mereghetti, B. Palano. Some formal tools for analyzing quantum automata. *In Theoretical Computer Science* **356**, pp. 14–25 (2006).

[13] A. Bertoni, B. Palano. *Linguaggi formali e automi.* Dispense per il corso di "Linguaggi formali e automi" per il CDL in Informatica, Università degli Studi di Milano.

[14] M.P. Bianchi, C. Mereghetti, B. Palano, G. Pighizzini. Probabilistic vs. nondeterministic unary automata. *In Proc. of the 2nd Workshop on Non-Classical Models of Automata and Applications (NCMA 2010), Jena, Germany,* pp. 33–44 (2010).

[15] M.P. Bianchi, C. Mereghetti, B. Palano, G. Pighizzini. On the Size of Unary Probabilistic and Nondeterministic Automata. *Fundamenta Informaticae* **112**, pp. 119–135 (2011).

[16] M.P. Bianchi, B. Palano. Events and languages on unary quantum automata. *In Proc. of the 1st Workshop on Non-Classical Models of Automata and Applications (NCMA 2009), Wroclaw, Poland,* pp. 61–75 (2009).

[17] M.P. Bianchi, B. Palano. Behaviours of unary quantum automata. *In Fundamenta Informaticae* **104**, pp. 1–15 (2010).

[18] M.P. Bianchi, G. Pighizzini. Normal forms for unary probabilistic automata. *In Proc. of the 3rd Workshop on Non-Classical Models of Automata and Applications, Milan, Italy,* pp. 9–102 (2011).

[19] M.P. Bianchi, G. Pighizzini. Normal forms for unary probabilistic automata. *In RAIRO, DOI: 10.1051/ita/2012017* (2012).

[20] V.D. Blondel, E. Jeandel, P. Koiran, N. Portier. Decidable and undecidable problems about quantum automata. *SIAM J. Comput.* **34**, pp. 1464–1473 (2005).

[21] A. Brodsky, N. Pippinger. Characterizations of 1-way quantum finite automata. *SIAM J. Comput.* **5**, pp. 1456–1478 (2002).

[22] M. Chrobak. Finite Automata and Unary Languages. *Theoretical Computer Science* **47(3)**, pp. 149–158 (1986).

[23] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. Roy. Soc. London Ser. A* 400, pp. 97–117 (1985).

[24] A. di Pierro. *Appunti dalle lezioni di Quantum Computing.* Lecture notes available at http://www.di.unipi.it/ dipierro/Didattica/QuantCom/. (In italian)

[25] R. Feynman. Simulating physics with computers. *Int. J. Theoretical Physics* **21**, pp. 467–488 (1982).

[26] V. Geffert. Magic numbers in the state hierarchy of finite automata. *Information and Computation* **205**, pp. 1652–1670 (2007).

[27] G. Gramlich. Probabilistic and Nondeterministic Unary Automata. *MFCS, LNCS 2757*, pp. 460-469 (2003).

[28] D.J. Griffiths. *Introduction to Quantum Mechanics. (2nd ed.)* Prentice Hall (2004).

[29] J. Gruska. *Quantum Computing.* McGraw-Hill, New York (1999).

[30] J. Gruska. Descriptional complexity issues in quantum computing. *J. Automata, Languages and Combinatorics* **5**, pp. 191–218, (2000).

[31] W. Höffdings. Probability inequalities for sums of bounded random variables. *J. American Statistical Association* **58**, pp. 13–30 (1963).

[32] M. Holzer, M. Kutrib. Descriptional complexity – an introductory survey. *In: Martin-Vide, C. (ed.) Scientific Applications of Language Methods*, pp. 1–58. Imperial College Press, London (2010).

[33] J.E. Hopcroft, R. Motwani, J.D. Ullman. *Introduction to automata theory, languages, and computation.* Addison-Wesley (2003).

[34] R.I.G. Hughes. *The Structure and Interpretation of Quantum Mechanics.* Harvard Univ. Press (1989).

[35] T. Jiang, E. McDowell, B. Ravikumar. The structure and complexity of minimal NFA's over a unary alphabet. *Int. J. Found. Comp. Sc.* **2**, pp. 163–182 (1991).

[36] J. Kaneps. Regularity of One-Letter Languages Acceptable by 2-Way Finite Probabilistic Automata. *FCT 1991, LNCS 529,* pp. 287–296 (1991).

[37] A. Kondacs, J. Watrous. On the Power of Quantum Finite State Automata. *38th Annual Symposium on Foundations of Computer Science*, pp. 66–75 (1997).

[38] F. Mera, G. Pighizzini. Complementing unary nondeterministic automata. *Theoretical Computer Science* **330**, pp. 349–360 (2005).

[39] C. Mereghetti, B. Palano. Quantum finite automata with control language. *In Theoretical Informatics and Applications* **40**, pp. 315–332 (2006).

[40] C. Mereghetti, B. Palano. Quantum automata for some multiperiodic languages. *Theoretical Computer Science* **387(2)**, pp. 177–186 (2007).

[41] C. Mereghetti, B. Palano, G. Pighizzini. Note on the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata. *Theoretical Informatics and Applications* **35**, pp. 477–490 (2001).

[42] C. Mereghetti, G. Pighizzini. Two-way automata simulations and unary languages. *J. Automata, Languages and Combinatorics* **5**, pp. 287–300 (2000).

[43] M. Milani, G. Pighizzini. Tight bounds on the simulation of unary probabilistic automata by deterministic automata. *J. Automata, Languages and Combinatorics* **6**, pp. 481–492 (2001).

[44] G.E. Moore. Progress in Digital Integrated Electronics. *In Digest of the 1975 International Electron Devices Meeting, IEEE,* pp. 11–13 (1975).

[45] C. Moore, J. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science* **237**, pp. 275–306 (2000).

[46] I. Niven. *Irrational numbers* - Carus Monographs, Vol. 11, The Mathematical Association of America, distributed by John Wiley and Sons (1956).

[47] A. Paz. *Introduction to Probabilistic Automata.* Academic Press, New York, London (1971).

[48] J.E. Pin. On languages accepted by finite reversible automata. *In Proc. 14th Int. Coll. Automata, Lang. and Prog. LNCS* **267**, Springer-Verlag, pp. 237–249 (1987).

[49] M.O. Rabin. Probabilistic automata. *Information and Control* **6**, pp. 230–245 (1963).

[50] M. O. Rabin, D. Scott. Finite automata and their decision problems. *IBM J. Res. Dev.* **3(2)**, pp. 114–125 (1959).

[51] E. Seneta. *Non–negative Matrices and Markov Chains. (2nd ed.)* Springer–Verlag (1981).

[52] G. Shilov. *Linear Algebra.* Prentice Hall, 1971. Reprinted by Dover, 1977.

[53] M. Szalay. On the maximal order in S_n and S_n^*. *Acta Arithm.* **37**, pp. 321–331 (1980).