Latvian Quantum Finite State Automata for Unary Languages

Carlo Mereghetti Beatrice Palano Priscilla Raucci

Dipartimento di Informatica “Giovanni Degli Antoni”
Università degli Studi di Milano, via Celoria 18, 20135 Milano – Italy
{carlo.mereghetti, beatrice.palano, priscilla.raucci}@unimi.it

We design *Latvian quantum finite state automata* (LQFAs for short) recognizing unary regular languages with isolated cut point $\frac{1}{2}$. From an architectural point of view, we combine two LQFAs recognizing with isolated cut point, respectively, the finite part and the ultimately periodic part of any given unary regular language L. In both modules, we use a component addressed in the literature and here suitably adapted to the unary case, to discriminate strings on the basis of their length. The number of basis states and the isolation around the cut point of the resulting LQFA for L exponentially depends on the size of the minimal deterministic finite state automaton for L.

1 Introduction

Quantum finite automata (QFAs for short) represent a theoretical model for a quantum computer with finite memory [3, 4]. While we can hardly expect to see a full-featured quantum computer in the near future, small quantum components, modeled by QFAs, seem to be promising from a physical implementation viewpoint (see, e.g., [7, 15]).

Very roughly speaking, a QFA is obtained by imposing the quantum paradigm — superposition, unitary evolution, observation — to a classical finite state automaton. The state of the QFA can be seen as a linear combination of classical states, called superposition. The QFA steps from a superposition to the next one by a unitary (reversible) evolution. Superpositions can transfer the complexity of a computation from a large number of sequential steps to a large number of coherently superposed classical states (this phenomenon is sometimes referred as quantum parallelism). Along its computation, the QFA can be “observed”, i.e., some features, called observables, can be measured. From measuring an observable, an outcome is obtained with a certain probability and the current superposition irreversibly “collapses”, with the same probability, to a particular superposition (coherent with the observed outcome).

QFAs exhibit both advantages and disadvantages with respect to their classical (deterministic or probabilistic) counterpart. Basically, quantum superposition offers some computational advantages on probabilistic superposition. On the other hand, quantum dynamics are reversible: because of limitation of memory, it is sometimes impossible to simulate deterministic finite state automata (DFAs for short) by quantum automata. Limitations due to reversibility can be partially attenuated by systematically introducing measurements of suitable observables as computational steps.

In the literature, several models of QFAs are proposed, which mainly differ in their measurement policy. The first and most simple model is the *measure-once* QFA (MO-QFA for short) [6, 16], where the probability of accepting strings is evaluated by “observing” just once, at the end of input processing. In *measure-many* QFAs (MM-QFAs for short) [11], instead, the acceptance probability is evaluated by observing after each move, thus allowing the possibility of halting the computation in the middle of input processing. An additional model is the *Latvian* QFA (LQFA for short) [1], which can be regarded as

Rudolf Freund, Benedek Nagy (Eds.): 13th International Workshop on
Non-Classical Models of Automata and Applications (NCMA 2023)
EPTCS 388, 2023, pp. 63–78, doi:10.4204/EPTCS.388.8

“intermediate” between MO-QFAs and MM-QFAs. In fact, as in the MM-QFA model, LQFAs are observed after each move; on the other hand, as in the MO-QFA model, acceptance probability is evaluated at the end of the computation only. From a language recognition point of view, it is well known that MO-QFAs are strictly less powerful than LQFAs, which are strictly less powerful than MM-QFAs, which are strictly less powerful than DFAs. This hierarchy is established, e.g., in [1, 6, 11].

In this paper, we investigate the architecture and size of LQFAs processing *unary languages*, i.e. languages built over a single-letter alphabet. A similar investigation is presented in [5], where MM-QFAs recognizing unary regular languages with isolated cut point are exhibited, whose size (number of basis states) is linear in the size of equivalent minimal DFAs. Here, we show that unary regular languages can be recognized with isolated cut point by the less powerful model of LQFAs as well, paying by an exponential size increase. A relevant module in our construction is a LQFA recognizing with isolated cut point the strings of length exceeding a fixed threshold. For its design, we adapt a construction in [1, 14] to the unary case. Such a module is then suitably combined with two LQFAs taking care, respectively, of the finite part and the ultimately periodic part any unary regular language consists of. The architecture of the resulting LQFA turns out to be significantly different from the equivalent MM-QFAs in [5]. Moreover, while in the MM-QFA case the isolation around the cut point is constant, for LQFAs it exponentially decreases with respect to the size of the DFA for the finite part of the target unary regular language. However, it should be stressed that the less powerful model of MO-QFAs cannot recognize with isolated cut point all unary regular languages. Our results constructively prove that LQFAs and MM-QFAs have the same recognition power, whenever restricted to recognize unary languages with isolated cut point.

The paper is organized as follows. In Section 2, we overview basics on formal language theory, linear algebra, and quantum finite state automaton models. In Section 3, we design isolated cut point LQFAs recognizing the strings whose length is greater than or equal to a fixed value. Then, in Section 4, we provide the full architecture of isolated cut point LQFAs for unary regular languages, analyzing their size, cut point, and isolation. Finally, in Section 5, we draw some concluding remarks and offer possible research hints.

2 Preliminaries

2.1 Formal Languages

We assume familiarity with basic notions of formal language theory (see, e.g., [9]). Given a set S, we let $|S|$ denote its cardinality. The set of all words or strings (including the empty string ε) over a finite alphabet Σ is denoted by Σ^*, and we let $\Sigma^+ = \Sigma^* \setminus \varepsilon$. For a string $w \in \Sigma^*$, we let $|w|$ denote its length and w_i its ith symbol. For any given $i \geq 0$, we let Σ^i be the set of strings over Σ of length i, with $\Sigma^0 = \{\varepsilon\}$. We let $\Sigma^{\leq i} = \bigcup_{j=0}^{i} \Sigma^j$; sets $\Sigma^{>i}$ and $\Sigma^{\geq i}$ are defined accordingly. A language over Σ is any subset $L \subseteq \Sigma^*$; its complement is the language $L^c = \Sigma^* \setminus L$. A *deterministic finite state automaton* (DFA) is formally defined as a 5-tuple $D = (Q, \Sigma, q_0, \delta, F)$, where Q is the finite set of states, Σ the finite input alphabet, $q_0 \in Q$ the initial state, $F \subseteq Q$ the set of accepting states, and $\delta : Q \times \Sigma \to Q$ the transition function. Denoting by δ^* the canonical extension of δ to Σ^*, the language recognized by D is the set $L_D = \{w \in \Sigma^* | \delta^*(q_0, w) \in F\}$. It is well known that DFAs characterize the class of regular languages.

A *unary language* is any language built over a single letter alphabet, e.g., $\Sigma = \{\sigma\}$, and thus has the general form $L \subseteq \sigma^*$. Unary *regular* languages form *ultimately periodic sets*, as stated by the following

Theorem 1. ([9, 17]) *Let $L \subseteq \sigma^*$ be a unary regular language. Then, there exist two integers $T \geq 0$ and $P > 0$ such that, for any $k \geq T$, we have $\sigma^k \in L$ if and only if $\sigma^{k+P} \in L$.*

By Theorem 1, it is easy to see that any unary regular language L can be recognized by a (minimal) DFA consisting of an initial path of T states joined to a cycle of P states; accepting states are suitably settled on both the path and the cycle. Unary regular languages satisfying Theorem 1 with $T = 0$ are called *periodic languages* of period P.

2.2 Linear Algebra

We quickly recall some notions of linear algebra, useful to describe quantum computational devices. For more details, we refer the reader to, e.g., [19]. The fields of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. Given a complex number $z = a + ib$, with $a, b \in \mathbb{R}$, its *conjugate* is denoted by $z^* = a - ib$, and its *modulus* by $|z| = \sqrt{z \cdot z^*}$. We let $\mathbb{C}^{n\times m}$ denote the set of $n \times m$ matrices with entries in $\mathbb{C}$. Given a matrix $M \in \mathbb{C}^{n\times m}$, for $1 \leq i \leq n$ and $1 \leq j \leq m$, we denote by M_{ij} its (i, j)th entry. The *transpose* of M is the matrix $M^T \in \mathbb{C}^{m\times n}$ satisfying ${M^T}_{ij} = M_{ji}$, while we let M^* be the matrix satisfying ${M^*}_{ij} = (M_{ij})^*$. The *adjoint* of M is the matrix $M^\dagger = (M^T)^*$. For matrices $A, B \in \mathbb{C}^{n\times m}$, their *sum* is the $n \times m$ matrix $(A + B)_{ij} = A_{ij} + B_{ij}$. For matrices $C \in \mathbb{C}^{n\times m}$ and $D \in \mathbb{C}^{m\times r}$, their *product* is the $n \times r$ matrix $(C \cdot D)_{ij} = \sum_{k=1}^{m} C_{ik} \cdot D_{kj}$. For matrices $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{p\times q}$, their *direct (or tensor or Kronecker) product* is the $n{\cdot}p \times m{\cdot}q$ matrix defined as

$$A \otimes B = \begin{pmatrix} A_{11}B & \cdots & A_{1m}B \\ \vdots & \ddots & \vdots \\ A_{n1}B & \cdots & A_{nm}B \end{pmatrix}.$$

When operations are allowed by matrix dimensions, we have that $(A \otimes B) \cdot (C \otimes D) = A \cdot C \, \otimes \, B \cdot D$.

A *Hilbert space* of dimension n is the linear space $\mathbb{C}^n$ of n-dimensional complex row vectors equipped with sum and product by elements in $\mathbb{C}$, where the *inner product* $\langle \varphi, \psi \rangle = \varphi \cdot \psi^\dagger$ is defined, for vectors $\varphi, \psi \in \mathbb{C}^n$. The ith component of φ is denoted by φ_i, its *norm* is given by $\|\varphi\| = \sqrt{\langle \varphi, \varphi \rangle} = \sqrt{\sum_{i=1}^{n} |\varphi_i|^2}$. If $\langle \varphi, \psi \rangle = 0$ (and $\|\varphi\| = 1 = \|\psi\|$), then φ and ψ are orthogonal (orthonormal). An orthonormal basis of $\mathbb{C}^n$ is any set of n orthonormal vectors in $\mathbb{C}^n$. In particular, the *canonical basis* of $\mathbb{C}^n$ is the set $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n\}$, where $\boldsymbol{e}_i \in \mathbb{C}^n$ is the vector having 1 at the ith component and 0 elsewhere. Clearly, any vector $\varphi \in \mathbb{C}^n$ can be univocally expressed as a linear combination of the vectors in the canonical basis as $\varphi = \sum_{i=1}^{n} \varphi_i \cdot \boldsymbol{e}_i$. This latter fact is usually addressed by saying that $\mathbb{C}^n$ is *spanned* by $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n\}$. Two subspaces $X, Y \subseteq \mathbb{C}^n$ are orthogonal if any vector in X is orthogonal to any vector in Y. In this case, we denote by $X \dot{+} Y$ the linear space generated by $X \cup Y$. For vectors $\varphi \in \mathbb{C}^n$ and $\psi \in \mathbb{C}^m$, their direct (or tensor or Kronecker) product is the vector $\varphi \otimes \psi = (\varphi_1 \cdot \psi, \ldots, \varphi_n \cdot \psi)$; we have $\|\varphi \otimes \psi\| = \|\varphi\| \cdot \|\psi\|$.

A matrix $M \in \mathbb{C}^{n\times n}$ is said to be *unitary* if $M \cdot M^\dagger = I^{(n)} = M^\dagger \cdot M$, where $I^{(n)}$ is the $n \times n$ identity matrix. Equivalently, M is unitary if it preserves the norm, i.e., $\|\varphi \cdot M\| = \|\varphi\|$ for any vector $\varphi \in \mathbb{C}^n$. Direct products of unitary matrices are unitary as well. The matrix M is said to be *Hermitian (or self-adjoint)* if $M = M^\dagger$. Let $\mathscr{O} \in \mathbb{C}^{n\times n}$ be an Hermitian matrix, $\nu_1, \nu_2, \ldots, \nu_s$ its eigenvalues, and $E_1, E_2, \ldots, E_s$ the corresponding eigenspaces. It is well known that each eigenvalue ν_k is real, that E_i is orthogonal to E_j for every $1 \leq i \neq j \leq s$, and that $E_1 \dot{+} E_2 \dot{+} \cdots \dot{+} E_s = \mathbb{C}^n$. Thus, every vector $\varphi \in \mathbb{C}^n$ can be uniquely decomposed as $\varphi = \varphi_{(1)} + \varphi_{(2)} + \cdots + \varphi_{(s)}$, for unique $\varphi_{(j)} \in E_j$. The linear transformation $\varphi \mapsto \varphi_{(j)}$ is the *projector* P_j onto the subspace E_j. Actually, the Hermitian matrix $\mathscr{O}$ is biunivocally determined by its eigenvalues and projectors as $\mathscr{O} = \nu_1 \cdot P_1 + \nu_2 \cdot P_2 + \cdots + \nu_s \cdot P_s$. We recall that a matrix $P \in \mathbb{C}^{n\times n}$ is a projector if and only if P is Hermitian and idempotent, i.e. $P^2 = P$.

Let $\omega = \mathrm{e}^{i\frac{2\pi}{n}}$ be the nth root of the unity ($\omega^n = 1$) and define the Vandermonde matrix $W \in \mathbb{C}^{n\times n}$ whose (r, c)th component is ω^{rc}, for $0 \leq r, c < n$. Let the $n \times n$ complex matrix $F_n = \frac{1}{\sqrt{n}} \cdot W$. It is easy to see that F_n is the unitary matrix implementing the *quantum Fourier transform*. Throughout the paper, it will be useful to recall that operating F_n on the jth vector of the canonical basis yields the vector $\boldsymbol{e}_j \cdot F_n = \frac{1}{\sqrt{n}} \cdot (\omega^0, \omega^{(j-1)\cdot 1}, \ldots, \omega^{(j-1)\cdot(n-1)})$. We remark that $|(\boldsymbol{e}_j \cdot F_n)_k|^2 = \frac{1}{n}$, for every $1 \le k \le n$.

As we will see in the next section, in accordance with quantum mechanics principles (see, e.g., [10]), the state of a quantum finite state automaton at any given time during its computation is represented by a norm 1 vector from a suitable Hilbert space, the state evolution of the automaton is modeled by unitary matrices, and information on certain characteristics of the automaton are probabilistically extracted by measuring some “observables” represented by Hermitian matrices.

2.3 Quantum Finite State Automata

Here, we recall the model of a *Latvian quantum finite state automaton* [1] we are mostly interested in. We then quickly introduce *measure-once* quantum finite state automata [6, 16] as a particular case of Latvian automata. Finally, we overview *measure-many* quantum finite state automata [11].

Definition 1. *Let Σ be an input alphabet, $\sharp \notin \Sigma$ an endmarker symbol, and set $\Gamma = \Sigma \cup \{\sharp\}$. A* Latvian quantum finite automaton (LQFA for short) *is a system $\mathscr{A} = (Q, \Sigma, \pi_0, \{U(\sigma)\}_{\sigma\in\Gamma}, \{\mathscr{O}_\sigma\}_{\sigma\in\Gamma}, Q_{acc})$, where*

- *$Q = \{q_1, q_2, \ldots, q_n\}$ is the finite set of basis states; the elements of Q span*[1] *the Hilbert space $\mathbb{C}^n$,*
- *$Q_{acc} \subseteq Q$ is the set of accepting basis states,*
- *$\pi_0 \in \mathbb{C}^n$ is the initial amplitude vector (superposition) satisfying $\|\pi_0\| = 1$,*
- *$U(\sigma) \in \mathbb{C}^{n\times n}$ is the unitary evolution matrix, for any $\sigma \in \Gamma$,*
- *for any $\sigma \in \Sigma$, we let $\mathscr{O}_\sigma = \sum_{i=0}^{k_\sigma - 1} c_i(\sigma) \cdot P_i(\sigma)$ be an observable (Hermitian matrix) on $\mathbb{C}^n$, where $\{c_0(\sigma), \ldots, c_{k_\sigma-1}(\sigma)\}$ is the set of all possible outcomes (eigenvalues) of measuring $\mathscr{O}_\sigma$, and $\{P_0(\sigma), \ldots, P_{k_\sigma-1}(\sigma)\}$ are the projectors onto the corresponding eigenspaces,*
- *we let $\mathscr{O}_\sharp = a \cdot P_{acc}(\sharp) + r \cdot (I^{(n)} - P_{acc}(\sharp))$ be the final observable, where $P_{acc}(\sharp)$ is the projector onto the subspace of $\mathbb{C}^n$ spanned by the states in Q_{acc}.*

Let us briefly describe the behavior of $\mathscr{A}$ on an input word $w\sharp \in \Sigma^*\sharp$. At any given time, the *state* of $\mathscr{A}$ is a *superposition of basis states* in Q which is represented by a norm 1 vector $\xi \in \mathbb{C}^n$. We have that $\xi_i \in \mathbb{C}$ is the *amplitude* of the basis state q_i, while $|\xi_i|^2 \in [0,1]$ is the *probability* of observing $\mathscr{A}$ in the basis state q_i. The computation of $\mathscr{A}$ on $w\sharp$ starts in the initial superposition π_0 by reading the first input symbol. Then, the transformations associated with each input symbol are applied in succession. The transformation corresponding to a symbol $\sigma \in \Gamma$ consists of two steps:

1. *Evolution:* the matrix $U(\sigma)$ acts on the current state ξ of $\mathscr{A}$, yielding the next state $\xi' = \xi \cdot U(\sigma)$.
2. *Observation:* the observable $\mathscr{O}_\sigma$ is measured and the outcome $c_i(\sigma)$ is seen with probability $\|\xi' \cdot P_i(\sigma)\|^2$; upon seeing $c_i(\sigma)$, according to Copenhagen interpretation of quantum mechanics [10], the state of $\mathscr{A}$ "collapses" to (norm 1) state $\xi' \cdot P_i(\sigma) / \|\xi' \cdot P_i(\sigma)\|$ and the computation continues, unless we are processing the endmarker $\sharp$.

Upon processing the endmarker $\sharp$, the final observable $\mathscr{O}_\sharp$ is measured yielding the probability of seeing $\mathscr{A}$ in an accepting basis state. Therefore, the probability of accepting $w \in \Sigma^*$ is given by

$$p_{\mathscr{A}}(w) = \sum_{i_1=0}^{k_{w_1}-1} \cdots \sum_{i_{|w|}=0}^{k_{w_{|w|}}-1} \left\| \pi_0 \cdot U(w_1) \cdot P_{i_1}(w_1) \cdot \cdots \cdot U(w_{|w|}) \cdot P_{i_{|w|}}(w_{|w|}) \cdot U(\sharp) \cdot P_{acc}(\sharp) \right\|^2 .$$

[1] We can associate with the set $Q = \{q_1, q_2, \ldots, q_n\}$ of basis states the canonical basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ of the Hilbert space $\mathbb{C}^n$ (see Section 2.2) where, for each $1 \le i \le n$, we let $\boldsymbol{e}_i$ represent the basis state q_i. As the canonical basis spans $\mathbb{C}^n$, with a slight abuse of notation, we say that the elements of Q spans $\mathbb{C}^n$.

The function $p_{\mathscr{A}} : \Sigma^* \to [0,1]$ is the *stochastic event induced by* $\mathscr{A}$. The *language recognized by* $\mathscr{A}$ *with cut point* $\lambda \in [0,1]$ is the set of words $L_{\mathscr{A},\lambda} = \{w \in \Sigma^* \mid p_{\mathscr{A}}(w) > \lambda\}$. The cut point λ is said to be *isolated* whenever there exists $\rho \in \left(0, \frac{1}{2}\right]$ such that $|p_{\mathscr{A}}(w) - \lambda| \geq \rho$, for any $w \in \Sigma^*$. The parameter ρ is usually referred to as *radius of isolation*.

In general, a language $L \subseteq \Sigma^*$ is recognized with isolated cut point by a LQFA whenever there exists a LQFA $\mathscr{A}$ such that $(\inf\{p_{\mathscr{A}}(w) \mid w \in L\} - \sup\{p_{\mathscr{A}}(w) \mid w \notin L\}) > 0$. In this case, we can compute the cut point as being $\lambda = \frac{1}{2} \cdot (\inf\{p_{\mathscr{A}}(w) \mid w \in L\} + \sup\{p_{\mathscr{A}}(w) \mid w \notin L\})$, with radius of isolation $\rho = \frac{1}{2} \cdot (\inf\{p_{\mathscr{A}}(w) \mid w \in L\} - \sup\{p_{\mathscr{A}}(w) \mid w \notin L\})$. Throughout the rest of the paper, for the sake of conciseness, we will sometimes be writing "isolated cut point quantum finite automaton for a language" instead of "quantum finite automaton recognizing a language with isolated cut point". Isolated cut point turns out to be one of the main language recognition policies within the literature of probabilistic devices. Its relevance in the realm of finite state automata is due to the fact that we can arbitrarily reduce the classification error probability of an input word w by repeating a constant number of times (not depending on the length of w) its parsing and taking the majority of the answers. We refer the reader to ,e.g., [18, Sec. 5], where the notion of isolated cut point recognition is introduced and carefully analyzed.

One of the two original and most studied models of a quantum finite state automaton is the *measure-once* model (MO-QFA for short). An MO-QFA can be seen as a particular LQFA where, for any $\sigma \in \Sigma$, we have that $\mathscr{O}_\sigma = I^{(n)}$. Basically, this amounts to leave the computation of $\mathscr{A}$ undisturbed up to the final observation for acceptance. Thus, an MO-QFA can be formally and more succinctly written as $\mathscr{A} = (Q, \Sigma, \pi_0, \{U(\sigma)\}_{\sigma\in\Gamma}, Q_{acc})$. The probability of $\mathscr{A}$ accepting the word $w \in \Sigma^*$ now simplifies as

$$p_{\mathscr{A}}(w) = \left\|\pi_0 \cdot U(w_1) \cdot \cdots \cdot U(w_{|w|}) \cdot U(\sharp) \cdot P_{acc}(\sharp)\right\|^2.$$

Let us now switch to the other original model, namely a *measure-many quantum finite state automaton* (MM-QFA for short). Roughly speaking, an MM-QFA $\mathscr{A}$ is defined as LQFA but with the possibility of accepting/rejecting the input string *before* reaching the endmarker. More precisely, the set Q of the basis states of $\mathscr{A}$ can be partitioned into *halting states*, which can be either *accepting* or *rejecting*, and *non halting states*, also called *go* states, i.e., $Q = Q_{acc} \cup Q_{rej} \cup Q_{go}$. Following such a state partition, the sole observable $\mathscr{O} = a \cdot P_{acc} + r \cdot P_{rej} + g \cdot P_{go}$, whose projectors map onto the subspaces spanned by the corresponding basis states, is associated with *each* symbol in Γ. At each step, the observable $\mathscr{O}$ is measured and the computation of $\mathscr{A}$ continues (unless we are processing $\sharp$) only if the outcome g is seen. Instead, if the outcome a (r) is seen, then $\mathscr{A}$ halts and accepts (rejects). Formally, the MM-QFA $\mathscr{A}$ can be written as $\mathscr{A} = (Q, \Sigma, \pi_0, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathscr{O}, Q_{acc})$, and the probability of accepting the word $w\sharp = w_1 \cdots w_n w_{n+1}$ is

$$p_{\mathscr{A}}(w) = \textstyle\sum_{k=1}^{n+1} \|\pi_0 \cdot (\prod_{i=1}^{k-1} U(w_i) \cdot P_{go}) \cdot U(w_k) \cdot P_{acc}\|^2.$$

It is well known (see, e.g., [6, 16]) that the class of languages recognized by isolated cut point MO-QFAs coincides with the class of group languages. Notice that finite languages are not group languages, and hence they cannot be accepted by isolated cut point MO-QFAs. Isolated cut point LQFAs are proved in [1, 14] to be strictly more powerful than isolated cut point MO-QFAs, since their recognition power coincides with the class of block group languages. An equivalent characterization states that a language is recognized by an isolated cut point LQFA if and only if it belongs to the boolean closure of languages of the form $L_1 a_1 L_2 a_2 \cdots a_k L_{k+1}$, for $a_i \in \Sigma$, group language $L_i \subseteq \Sigma^*$, and $|\Sigma| > 1$. Finally, the recognition power of isolated cut point MM-QFAs still remains an open question. However, it is know that MM-QFAs are strictly more powerful than LQFAs but strictly less powerful than DFAs. In fact, isolated cut point MM-QFAs can recognize the language $a\Sigma^*$ which cannot be accepted by isolated cut point LQFAs [1]. On the other hand, isolated cut point MM-QFAs cannot recognize the language $\Sigma^* a$, for $|\Sigma| > 1$ and $a \in \Sigma$ [11].

3 Isolated Cut Point LQFAs for Words Longer than T

Here, we design an isolated cut point LQFA recognizing the unary language $\sigma^{\geq T}$, for any given $T > 0$ (i.e., the set of unary strings whose length is greater than or equal to T). As it will be clear in the next section, this LQFA will be a relevant component in the modular construction of isolated cut point LQFAs for unary regular languages.

Our design pattern is inspired by [1, 14], where the authors provide an isolated cut point LQFA for the language $\Sigma^* a_1 \Sigma^* a_2 \cdots a_k \Sigma^*$, with $a_i \in \Sigma$ and $|\Sigma| > 1$. So, we focus on recognizing the unary version of $\Sigma^* a_1 \Sigma^* a_2 \cdots a_T \Sigma^*$ yielded by fixing $a_1 = \cdots = a_T = \sigma$ and $\Sigma = \{\sigma\}$, namely, the desired language $a_1 \cdots a_T \Sigma^* = \sigma^{\geq T}$. We adapt the construction in [1, 14], and inductively exhibit a family $\{M^{(\ell)}\}_{\ell \geq 1}$ of LQFAs such that: *(i)* $M^{(\ell)}$ recognizes the language $\sigma^{\geq \ell}$ with isolated cut point, and *(ii)* $M^{(\ell)}$ is constructed by "expanding" $M^{(\ell-1)}$. So, the desired isolated cut point LQFA for $\sigma^{\geq T}$ will result after T "expansions", starting from the LQFA $M^{(1)}$. We provide a detailed analysis of the stochastic behavior of $M^{(\ell)}$ machines, emphasizing cut points, isolations and their size (i.e., number of their basis states). In this section, to have a convenient notation, we will be using A_σ for the evolution operator of our LQFAs.

Base of the construction: For the induction base, we define the LQFA $M^{(1)}$ for the language $\sigma^{\geq 1}$ as

$$M^{(1)} = (Q^{(1)}, \{\sigma\}, \pi_0, \{A_\sigma^{(1)}, A_\sharp^{(1)}\}, \{\mathscr{O}_\sigma^{(1)}, \mathscr{O}_\sharp^{(1)}\}, Q_{acc}^{(1)}),$$

where $Q^{(1)} = \{q_0, \ldots, q_{n-1}\}$ is the set of n basis states, $\pi_0 = \boldsymbol{e}_1$ meaning that $M^{(1)}$ starts in the state q_0, $Q_{acc}^{(1)} = Q^{(1)} \setminus \{q_0\}$ is the set of $n-1$ accepting states. For the evolution matrices, we let $A_\sigma^{(1)} = F_n$ (the quantum Fourier transform) and $A_\sharp^{(1)} = I$ (the identity matrix). The observable $\mathscr{O}_\sigma^{(1)}$ is the *canonical observable* defined by the projectors $\{\boldsymbol{e}_1{}^\dagger \cdot \boldsymbol{e}_1,\ \boldsymbol{e}_2{}^\dagger \cdot \boldsymbol{e}_2,\ \ldots, \boldsymbol{e}_n{}^\dagger \cdot \boldsymbol{e}_n\}$. By measuring $\mathscr{O}_\sigma^{(1)}$ on $M^{(1)}$ being in the superposition $\xi \in \mathbb{C}^n$, we will see $M^{(1)}$ in the basis state q_{i-1} with probability $\|\xi \cdot (\boldsymbol{e}_i{}^\dagger \cdot \boldsymbol{e}_i)\|^2 = |\xi_i|^2$. Upon such an outcome, the state of $M^{(1)}$ clearly collapses to $\boldsymbol{e}_i$. The final observation $\mathscr{O}_\sharp^{(1)}$ projects onto the subspace spanned by the accepting basis states $\{q_1, \ldots, q_{n-1}\}$.

The automaton $M^{(1)}$ behaves as follows: when the first input symbol is read, the state of $M^{(1)}$ becomes $\pi_0 \cdot A_\sigma^{(1)} = \boldsymbol{e}_1 \cdot F_n$, upon which the canonical observation is measured. As noticed at the end of Section 2.2, such a measurement will cause $M^{(1)}$ to move from q_0 to some basis state q_i, with $0 \leq i \leq n-1$, uniformly at random (i.e., with probability $|(\boldsymbol{e}_1 \cdot F_n)_{i+1}|^2 = \frac{1}{n}$). After processing (again, by quantum Fourier transform followed by measuring the canonical observable) the next input symbol from being in the state $\boldsymbol{e}_i$, we again find $M^{(1)}$ in a basis state uniformly at random. Such a dynamics continues unaltered, until the endmarker is reached and processed by the identity matrix. At this point, the final observation $\mathscr{O}_\sharp$ is measured, and an accepting state is easily seen to be reached with probability $|Q_{acc}^{(1)}| \cdot \frac{1}{n} = (\frac{n-1}{n})$. Clearly, processing the empty string leaves $M^{(1)}$ in the non accepting state q_0 with certainty. Therefore, $p_{M^{(1)}}(\varepsilon) = 0$, while for $k > 0$ we have $p_{M^{(1)}}(\sigma^k) = (\frac{n-1}{n})$. So, $M^{(1)}$ recognizes the language $\sigma^{\geq 1}$ with isolated cut point.

Inductive step of the construction: For the inductive step, we show how to build the isolated cut point LQFA $M^{(\ell)}$ for the language $\sigma^{\geq \ell}$ from the LQFA $M^{(\ell-1)}$ for the language $\sigma^{\geq \ell-1}$, this latter LQFA being given by inductive hypothesis. We define

$$M^{(\ell)} = (Q^{(\ell)}, \{\sigma\}, \pi_0, \{A_\sigma^{(\ell)}, A_\sharp^{(\ell)}\}, \{\mathscr{O}_\sigma^{(\ell)}, \mathscr{O}_\sharp^{(\ell)}\}, Q_{acc}^{(\ell)}),$$

where the set $Q^{(\ell)}$ of basis states consists of the previous set $Q^{(\ell-1)}$ of basis states, plus $(n-1)$ new basis states per each state in $Q_{acc}^{(\ell-1)}$. We let $Q_{acc}^{(\ell)}$ be the set containing these $(n-1) \cdot |Q_{acc}^{(\ell-1)}|$ new states, with $|Q_{acc}^{(\ell)}| = (n-1)^\ell$. Therefore, $Q^{(\ell)} = Q^{(\ell-1)} \cup Q_{acc}^{(\ell)} = \{q_0\} \cup Q_{acc}^{(1)} \cup Q_{acc}^{(2)} \cup \cdots \cup Q_{acc}^{(\ell)}$ with $|Q_{acc}^{(i)}| = (n-1)^i$, so that $|Q^{(\ell)}| = \sum_{i=0}^{\ell}(n-1)^i = \frac{(n-1)^{(\ell+1)}-1}{n-2}$. The initial superposition is $\pi_0 = \boldsymbol{e}_1$. We let $A_{\sharp}^{(\ell)} = I$ and $A_{\sigma}^{(\ell)} = B^{(\ell)} \cdot \tilde{A}^{(\ell-1)}$, where $\tilde{A}^{(\ell-1)}$ is the transformation acting as $A_{\sigma}^{(\ell-1)}$ on $Q^{(\ell-1)} \subset Q^{(\ell)}$, and as the identity elsewhere. Instead, $B^{(\ell)}$ is an additional operator working as follows. For any $\tilde{q} \in Q_{acc}^{(\ell-1)}$, let $Q_{\tilde{q}} = \{\tilde{q}_1, \ldots, \tilde{q}_{n-1}\} \subset Q_{acc}^{(\ell)}$ be the set of the $n-1$ new added accepting states associated with $\tilde{q}$. Thus, the operator $B^{(\ell)}$ first acts as F_n on $\{\tilde{q}\} \cup Q_{\tilde{q}}$ for every $\tilde{q} \in Q_{acc}^{(\ell-1)}$, then it measures $\mathscr{O}_{\sigma}^{(\ell)}$ being the canonical observable on $Q_{acc}^{(\ell-1)} \cup Q_{acc}^{(\ell)}$ plus the identity projector on the remaining basis states. The final observable $\mathscr{O}_{\sharp}^{(\ell)}$ as usual projects onto the subspace spanned by $Q_{acc}^{(\ell)}$. Actually, the automaton so far constructed does not perfectly comply with the definition of a LQFA given in Section 2.3 since $A_{\sigma}^{(\ell)}$ is not a unitary matrix. However, [1, Claim 1] ensures that the action of the operator $B^{(\ell)} \cdot \tilde{A}^{(\ell-1)}$ followed by measuring $\tilde{\mathscr{O}}_{\sigma}^{(\ell-1)}$ (the observable of $M^{(\ell-1)}$ extended to $Q^{(\ell)}$ by the identity projector onto $Q_{acc}^{(\ell)}$) can be expressed as a unitary matrix followed by measuring a suitable observable. This last detail possibly enlarges the dimension of the Hilbert space for $M^{(\ell)}$ by a factor bounded by n^{ℓ}. The stochastic event induced by $M^{(\ell)}$ will be discussed later.

To clarify the architecture and behavior of this family of automata, we now describe the LQFA $M^{(3)}$ recognizing the language $\sigma^{\geq 3}$ with isolated cut point. We have

$$M^{(3)} = (Q^{(3)}, \{\sigma\}, \pi_0, \{A_{\sigma}^{(3)}, A_{\sharp}^{(3)}\}, \{\mathscr{O}_{\sigma}^{(3)}, \mathscr{O}_{\sharp}^{(3)}\}, Q_{acc}^{(3)}),$$

where we let the set of basis states be $Q^{(3)} = \{q_0\} \cup Q_{acc}^{(1)} \cup Q_{acc}^{(2)} \cup Q_{acc}^{(3)}$ with $Q_{acc}^{(1)} = \{q_i \mid 1 \leq i \leq n-1\}$, $Q_{acc}^{(2)} = \{q_{i,j} \mid 1 \leq i,j \leq n-1\}$, and $Q_{acc}^{(3)} = \{q_{i,j,k} \mid 1 \leq i,j,k \leq n-1\}$. We remark that $Q_{acc}^{(3)}$ is the set of $(n-1)^3$ accepting basis states of $M^{(3)}$. We can regard basis states as partitioned into three groups reflected by the number of subscripts attributed to each basis state; each group of states is added in a subsequent step of the inductive construction. The general structure of the state (superposition) of $M^{(3)}$ is a norm 1 vector in $\mathbb{C}^{|Q^{(3)}|}$ of the following form, with $\alpha(q)$ denoting the amplitude of the basis state q:

$$\begin{aligned}
&[\alpha(q_0), \alpha(q_1), \alpha(q_{1,1}), [\ldots\, \alpha(q_{1,1,k})\, \ldots], \alpha(q_{1,2}), [\ldots\, \alpha(q_{1,2,k})\, \ldots], \;\ldots\;, \alpha(q_{1,n-1}), [\ldots\, \alpha(q_{1,n-1,k})\, \ldots],\\
&\alpha(q_2), \alpha(q_{2,1}), [\ldots\, \alpha(q_{2,1,k})\, \ldots], \alpha(q_{2,2}), [\ldots\, \alpha(q_{2,2,k})\, \ldots], \;\ldots\;, \alpha(q_{2,n-1}), [\ldots\, \alpha(q_{2,n-1,k})\, \ldots],\\
&\vdots\\
&\alpha(q_{n-1}), \alpha(q_{n-1,1}), [\ldots\, \alpha(q_{n-1,1,k})\, \ldots], \alpha(q_{n-1,2}), [\ldots\, \alpha(q_{n-1,2,k})\, \ldots], \;\ldots\;, \alpha(q_{n-1,n-1}), [\ldots\, \alpha(q_{n-1,n-1,k})\, \ldots]].
\end{aligned}$$

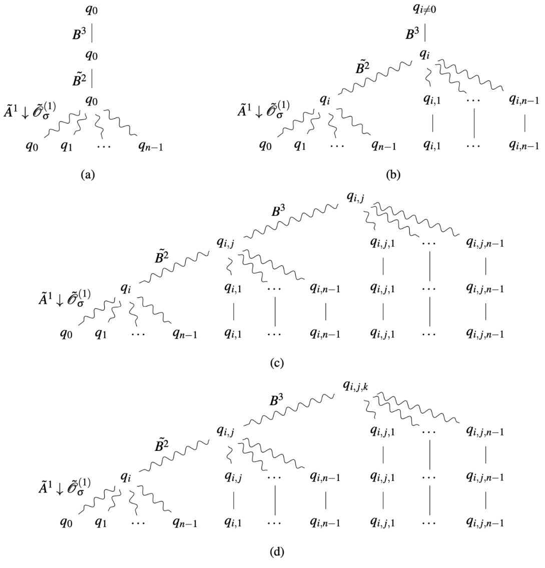
(*) Form of states (superpositions) of $M^{(3)}$.

As usual, we let $\pi_0 = \boldsymbol{e}_1$. The evolution matrices of $M^{(3)}$ are $A_{\sharp}^{(3)} = I$, while we have $A_{\sigma}^{(3)} = B^{(3)} \cdot \tilde{B}^{(2)} \cdot \tilde{A}^{(1)}$, where each matrix in the product acts on levels of the basis states as follows: $\tilde{A}^{(1)}$ affects the states in $\{q_0\} \cup Q_{acc}^{(1)}$, $\tilde{B}^{(2)}$ the states in $Q_{acc}^{(1)} \cup Q_{acc}^{(2)}$, and $B^{(3)}$ the states in $Q_{acc}^{(2)} \cup Q_{acc}^{(3)}$. From now on, it will be useful to describe the dynamic of $M^{(3)}$ by displaying the sequence of the stochastic vectors obtained by squaring the amplitudes in the superpositions of the form in (*). In such vectors, the value $|\alpha(q)|^2$ of the component associated with q represents the probability for $M^{(3)}$ of being in the basis state q. This stochastic dynamic description turns out to be appropriate as $M^{(3)}$ uses the canonical observable after each quantum Fourier transform operation. Upon reading a symbol σ, the LQFA $M^{(3)}$ executes $A_{\sigma}^{(3)}$ followed by measuring $\tilde{\mathscr{O}}_{\sigma}^{(1)}$: formally, we write $A_{\sigma}^{(3)} \downarrow \tilde{\mathscr{O}}_{\sigma}^{(1)}$. This operation distributes the probability differently in the three group of basis states $Q^{(1)}$, $Q_{acc}^{(2)}$ and $Q_{acc}^{(3)}$. In particular, the probability values turn out to be identical within each group of basis states, for each step of computation (except for the initial superposition π_0). Therefore, the form of the stochastic vector at each step of computation is

$$\begin{aligned}
[x, x, y, [\cdots z \cdots], y, [\cdots z \cdots], \ldots, y, [\cdots z \cdots],\\
x, y, [\cdots z \cdots], y, [\cdots z \cdots], \ldots, y, [\cdots z \cdots],\\
\vdots\qquad\qquad\\
x, y, [\cdots z \cdots], y, [\cdots z \cdots], \ldots, y, [\cdots z \cdots]],
\end{aligned}$$

where x is the probability value for the states in $Q^{(1)}$, y for the states in $Q^{(2)}_{acc}$, and z for the (accepting) states in $Q^{(3)}_{acc}$. Thus, the current accepting probability is $(n-1)^3 \cdot z$.

Now, let $x(k)$, $y(k)$, and $z(k)$ be the above basis states probabilities after processing the kth input symbol. We are going to establish the dependence of such values from $x(k-1)$, $y(k-1)$, and $z(k-1)$ in order to single out a closed formula for the stochastic event $p_{M^{(3)}}$. To this aim, for reader's ease of mind, a graphical representation is given in Figure 1, of how one step of the evolution-plus-observation $A^{(3)}_\sigma \downarrow \tilde{\mathscr{O}}^{(1)}_\sigma$ affects the probability values in each different group of basis states.



q_0
B^3
q_0
$\tilde{B}^2$
$\tilde{A}^1 \downarrow \tilde{\mathscr{O}}^{(1)}_\sigma$
q_0
q_0 q_1 $\cdots$ q_{n-1}
(a)
$q_{i\neq 0}$
B^3
q_i
$\tilde{B}^2$
q_i
$q_{i,1}$ $\cdots$ $q_{i,n-1}$
q_0 q_1 $\cdots$ q_{n-1} $q_{i,1}$ $\cdots$ $q_{i,n-1}$
(b)
$q_{i,j}$
$q_{i,j,1}$ $\cdots$ $q_{i,j,n-1}$
$q_{i,1}$ $\cdots$ $q_{i,n-1}$
(c)
$q_{i,j,k}$
$q_{i,j}$ $\cdots$ $q_{i,n-1}$
(d)

Figure 1: Stochastic representation of a computation step of $M^{(3)}$ on the symbol σ for basis states of different groups. The notation $\tilde{A}^1 \downarrow \tilde{\mathscr{O}}^{(1)}_\sigma$ means that $\tilde{A}^1$ is applied and then the observable $\tilde{\mathscr{O}}^{(1)}_\sigma$ is measured. Wave (straight) edges indicate basis state transitions occurring with probability $\frac{1}{n}$ (with certainty). For instance, the tree in (b) says that, starting from $q_{i\neq 0}$ for a fixed i and after one step of computation, we will observe $M^{(3)}$ in q_0 with probability $\frac{1}{n^2}$. Note that there exist $n-1$ trees of the form (b) leading to q_0.

Let us focus, e.g., on $x(k)$. The probability $x(k)$ depends on $x(k-1)$, $y(k-1)$, and $z(k-1)$ as follows:

- Figure 1(a) shows that the basis state q_0 contributes with $\frac{1}{n}\cdot x(k-1)$.
- Figure 1(b) shows the contribution of each basis states in $Q_{acc}^{(1)}$, which is $\frac{1}{n^2}\cdot x(k-1)$; given that $|Q_{acc}^{(1)}|=(n-1)$, the total contribution is $\frac{(n-1)}{n^2}\cdot x(k-1)$.
- Figure 1(c) shows that the total contribution given by $y(k-1)$ elements (i.e., by the $(n-1)^2$ basis states in $Q_{acc}^{(2)}$) is $\frac{(n-1)^2}{n^3}\cdot y(k-1)$.
- Figure 1(d) shows that the total contribution given by $z(k-1)$ elements (i.e., by the $(n-1)^3$ basis states in $Q_{acc}^{(3)}$) is $\frac{(n-1)^3}{n^3}\cdot z(k-1)$.

By analogous reasonings, we can obtain recurrences for $y(k)$ and $z(k)$, globally yielding the system

$$\begin{cases} x(k)=\frac{1}{n}\cdot x(k-1)+\frac{(n-1)}{n^2}\cdot x(k-1)+\frac{(n-1)^2}{n^3}\cdot y(k-1)+\frac{(n-1)^3}{n^3}\cdot z(k-1)\\ y(k)=\frac{1}{n}\cdot x(k-1)+\frac{(n-1)}{n^2}\cdot y(k-1)+\frac{(n-1)^2}{n^2}\cdot z(k-1)\\ z(k)=\frac{1}{n}\cdot y(k-1)+\frac{(n-1)}{n}\cdot z(k-1).\end{cases} \tag{1}$$

The base for this system of recurrences is the probability distribution after reading the first symbol σ, i.e.:

$$\begin{cases} x(1)=\frac{1}{n}\\ y(1)=z(1)=0.\end{cases} \tag{2}$$

From the system (1), the reader may verify that at each computation step the probability "shifts" towards the next deeper level of the basis states until reaching the basis states in $Q_{acc}^{(3)}$. In fact, after the first step (yielding probabilities in (2)), only the x-components have non null values. After the second step, only the x- and y-components have values different from 0, while the value of the z-components is still 0. This shows that $M^{(3)}$ rejects with certainty the strings in $\sigma^{\leq 2}$. After the third step, all the components have non null values; in particular, $z(3)=\frac{1}{n^3}$, so that the accepting probability of the string σ^3 attains $|Q_{acc}^{(3)}|\cdot z(3)=(\frac{n-1}{n})^3$. By solving the system (1), we get a closed formula for $z(k)$, with $k\geq 2$, as

$$z(k)=\frac{1}{n(n-1)^2}\cdot\left(1-\frac{(\frac{2n-2}{n^2})^{k-2}\cdot(n-1)^2+1}{(n-1)^2+1}\right).$$

This allows us to evaluate the accepting probability of $M^{(3)}$ for any string in σ^* as

$$p_{M^{(3)}}(\sigma^k)=|Q_{acc}^{(3)}|\cdot z(k)=\begin{cases}0 & \text{if } k\leq 2\\ \frac{n-1}{n}\cdot\left(1-\frac{(\frac{2n-2}{n^2})^{k-2}\cdot(n-1)^2+1}{(n-1)^2+1}\right) & \text{if } k\geq 3.\end{cases} \tag{3}$$

Equation (3) shows that $M^{(3)}$ recognizes $\sigma^{\geq 3}$ with isolated cut point. Clearly, the stochastic event induced by $M^{(3)}$ depends on the number n of the basis states of $M^{(1)}$, the initial automaton of the inductive construction. Figure 2 displays $p_{M^{(3)}}$ for some values of n. As expected, the higher n grows, the better the isolation around the cut point becomes.

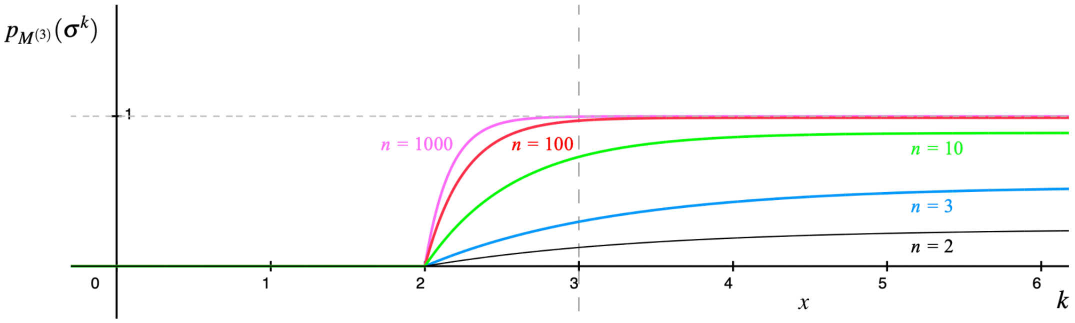


Figure 2: The ("continuous version" of the) stochastic events induced by $M^{(3)}$ according to Equation 3, for different values of the number n of basis states of $M^{(1)}$, the base module inductively leading to $M^{(3)}$.

Now, we consider the general LQFA $M^{(\ell)}$, and derive the system of recurrences for its stochastic dynamic. The set of basis states of $M^{(\ell)}$ is now partitioned into ℓ groups. For $1 \leq h \leq \ell$, we denote by $x_h(k)$ the probability for $M^{(\ell)}$ of being in a basis state of the hth group, after processing k input symbols. The system of recurrences for $M^{(\ell)}$ generalizes the system (1) as follows:

$$\begin{cases} x_1(k) = \frac{1}{n}\cdot x_1(k-1) + \sum_{j=1}^{\ell-1}\frac{(n-1)^j}{n^{j+1}}\cdot x_j(k-1) + \frac{(n-1)^\ell}{n^\ell}\cdot x_\ell(k-1) \\ x_2(k) = \left(\sum_{j=0}^{\ell-2}\frac{(n-1)^j}{n^{j+1}}\cdot x_{j+1}(k-1)\right) + \frac{(n-1)^{\ell-1}}{n^{\ell-1}}\cdot x_\ell(k-1) \\ \qquad\vdots \\ x_h(k) = \left(\sum_{j=0}^{\ell-h}\frac{(n-1)^j}{n^{j+1}}\cdot x_{j+h-1}(k-1)\right) + \frac{(n-1)^{\ell-h+1}}{n^{\ell-h+1}}\cdot x_\ell(k-1) \\ \qquad\vdots \\ x_\ell(k) = \frac{1}{n}\cdot x_{\ell-1}(k-1) + \frac{(n-1)}{n}\cdot x_\ell(k-1), \end{cases} \tag{4}$$

with initial values $x_1(1) = \frac{1}{n}$, and $x_h(1) = 0$ for every $2 \leq h \leq \ell$. We show the validity of this system of recurrences by induction, having, e.g., the system (1) for the automaton $M^{(3)}$ as base case. By inductive hypothesis, we assume the system of recurrences for $M^{(\ell-1)}$, and we build the system (4) for $M^{(\ell)}$. We consider the set of trees representing one step of the computation of our automata, starting from basis states of different groups. E.g., Figure 1 displays the four different types of trees for $M^{(3)}$, one per each group of basis states, plus one for the evolution from the state q_0. So, for $M^{(\ell)}$ we are going to provide ℓ of such trees, plus the one for q_0. Let us explain how to obtain them from the trees of $M^{(\ell-1)}$. Let $T_j^{(\ell-1)}$ be a tree representing one step of the evolution of $M^{(\ell-1)}$ on a basis state of group $1 \leq j \leq \ell-1$, namely, a basis state from $Q_{acc}^{(j)}$. Moreover, let $T_0^{(\ell-1)}$ be the tree for q_0. The evolution for $M^{(\ell)}$ is $A_\sigma^{(\ell)} = B^{(\ell)}\cdot\tilde{A}^{(\ell-1)}$. Thus, the behavior of $M^{(\ell)}$ is described by $\ell+1$ trees with the following structure:

- The trees $T_j^{(\ell)}$ for $0 \leq j < \ell-1$ are basically the trees $T_j^{(\ell-1)}$ with a preliminary step due to the action of $B^{(\ell)}$. Since in these trees the root is labeled by a basis state of level $j < \ell-1$, such a preliminary step coincides with the identity evolution.
- Even the trees $T_{\ell-1}^{(\ell)}$ and $T_\ell^{(\ell)}$ have the action of $B^{(\ell)}$ as a preliminary step. However, in these cases, $B^{(\ell)}$ acts as F_n on the basis states of groups $\ell-1$ in the tree $T_{\ell-1}^{(\ell)}$, and ℓ in the tree $T_\ell^{(\ell)}$. The structure of these two trees, both containing the tree $T_{\ell-1}^{\ell-1}$ as a sub-tree, is presented in Figure 3.

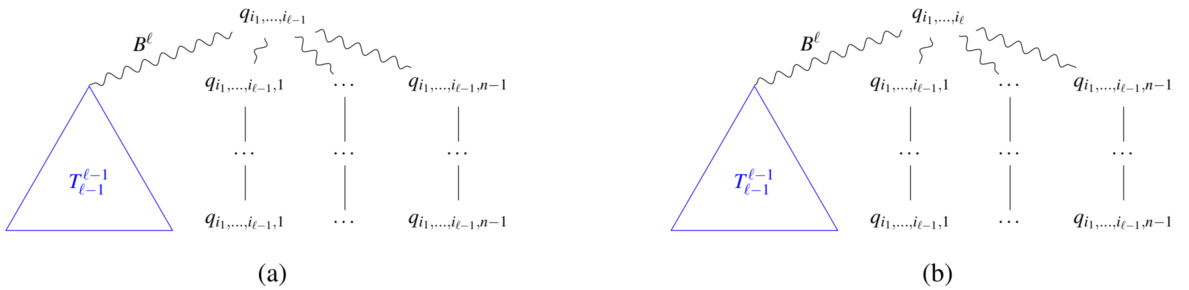

$q_{i_1,\ldots,i_{\ell-1}}$
B^ℓ
$T_{\ell-1}^{\ell-1}$
$q_{i_1,\ldots,i_{\ell-1},1}$
$q_{i_1,\ldots,i_{\ell-1},n-1}$
(a)
$q_{i_1,\ldots,i_\ell}$
B^ℓ
$T_{\ell-1}^{\ell-1}$
$q_{i_1,\ldots,i_{\ell-1},1}$
$q_{i_1,\ldots,i_{\ell-1},n-1}$
(b)

Figure 3: The form of the tree $T_{\ell-1}^{\ell}$ in (a), and of the tree T_{ℓ}^{ℓ} in (b) for the automaton $M^{(\ell)}$. Within both these two trees, the tree $T_{\ell-1}^{\ell-1}$ turns out to be a sub-tree.

It is now possible to properly justify the system (4) by using the induction step. Starting from the system of recurrences for $M^{(\ell-1)}$, we show how it modifies towards the system for $M^{(\ell)}$. Clearly, a new recurrence for $x_\ell(k)$ (i.e., the probabilities for basis states of group ℓ, the accepting states for $M^{(\ell)}$) is added at the end of the system. This component receives contributions only from the trees $T_{\ell-1}^{(\ell)}$ and $T_{\ell}^{(\ell)}$ weighted, respectively, by $x_{\ell-1}(k-1)$ and $x_\ell(k-1)$. Precisely, from the former tree we get the contribution $\frac{1}{n}\cdot x_{\ell-1}(k-1)$, from the latter ($n-1$ different trees) the contribution is $\frac{n-1}{n}\cdot x_\ell(k-1)$. For $x_h(k)$, with $1\leq h\leq \ell-1$, we note that the only modified contribution is the one carried by $x_{\ell-1}(k-1)$; moreover a new contribution from $x_\ell(k-1)$ is added. Even in this case, the trees $T_{\ell-1}^{(\ell)}$ and $T_{\ell}^{(\ell)}$ account for these modifications: the new coefficient of $x_{\ell-1}(k-1)$ is the old one for $x_{\ell-1}(k-1)$ (i.e., the one associated with $x_{\ell-1}(k-1)$ in the system for $M^{(\ell-1)}$) multiplied by $\frac{1}{n}$, while the coefficient of the new contribution $x_\ell(k-1)$ is the old one for $x_{\ell-1}(k-1)$ multiplied by $\frac{(n-1)}{n}$.

By simply applying repeated substitutions in the system (4), one may verify that, for $1\leq k\leq \ell$, the value $x_k(k)$ always equals $\frac{1}{n^k}$, while we have $x_{k+1}(k)=\cdots=x_\ell(k)=0$. Nevertheless, this implies that the acceptance probability of $M^{(\ell)}$ for the string σ^k is zero for $k<\ell$, while is $|Q_{acc}^{(\ell)}|\cdot x_\ell(\ell)=(\frac{n-1}{n})^\ell$ for $k=\ell$. We are now going to prove that for the strings in the language $\sigma^{\geq\ell}$ the acceptance probability never goes below $(\frac{n-1}{n})^\ell$. To this aim, it suffices to show

Theorem 2. *On the input string $\sigma^{\ell+s}$, with $s\geq 0$, the probability for $M^{(\ell)}$ of being in one of the accepting basis states in $Q_{acc}^{(\ell)}$ while processing the suffix σ^s is greater than or equal to $\frac{1}{n^\ell}$.*

Proof. We split the proof into two parts, both proved by induction. In the first part, we focus on the input prefix σ^ℓ. We show by induction on $1\leq k\leq \ell$ that $x_h(k)\geq\frac{1}{n^k}$ in the system (4) holds true for every $1\leq h\leq k$. This will enables us to obtain that $x_1(\ell),\ldots,x_\ell(\ell)\geq\frac{1}{n^\ell}$. For the base case $k=1$, we recall that $x_1(1)=\frac{1}{n}$. So, let us assume by inductive hypothesis that $x_h(k)\geq\frac{1}{n^k}$ for a given $k<\ell$ and every $1\leq h\leq k$, and prove the property for $k+1$. From the system (4), we have

$$x_h(k+1)=\frac{1}{n}\cdot x_{h-1}(k)+\frac{n-1}{n^2}\cdot x_h(k)+\frac{(n-1)^2}{n^3}\cdot x_{h+1}(k)+\cdots+\frac{(n-1)^{k-h+1}}{n^{k-h+2}}\cdot x_k(k)+\cdots+\frac{(n-1)^{\ell-h+1}}{n^{\ell-h+1}}\cdot x_\ell(k).$$

Since $x_h(k)\geq\frac{1}{n^k}$ for $1\leq h\leq k$, and 0 otherwise, we can bound $x_h(k+1)$ from below as

$$x_h(k+1)\geq\frac{1}{n^k}\cdot\frac{1}{n}\cdot\left(1+\frac{n-1}{n}+\frac{(n-1)^2}{n^2}+\cdots+\frac{(n-1)^{k-h+1}}{n^{k-h+1}}\right)\geq\frac{1}{n^{k+1}}.$$

Now, the second part of the proof comes, where we show, again by induction on k, that $x_h(k)\geq\frac{1}{n^\ell}$ for $k\geq\ell$ and $1\leq h\leq\ell$. By the first part of the proof, we have $x_1(\ell),x_2(\ell),\ldots x_\ell(\ell)\geq\frac{1}{n^\ell}$, and so the base case holds true. We prove $x_h(k+1) \geq \frac{1}{n^\ell}$ assuming such a property for k by inductive hypothesis. From the system (4), we get

$$x_h(k+1) = \frac{1}{n} \cdot x_{h-1}(k) + \frac{n-1}{n^2} \cdot x_h(k) + \frac{(n-1)^2}{n^3} \cdot x_{h+1}(k) + \ldots + \frac{(n-1)^{\ell-h}}{n^{\ell-h+1}} \cdot x_{\ell-1}(k) + \frac{(n-1)^{\ell-h+1}}{n^{\ell-h+1}} \cdot x_\ell(k).$$

Since we are assuming all $x_j(k)$'s to be greater than or equal to $\frac{1}{n^\ell}$, we can bound $x_h(k+1)$ from below as

$$x_h(k+1) \geq \frac{1}{n^\ell} \cdot \left(\frac{1}{n} + \frac{n-1}{n^2} + \frac{(n-1)^2}{n^3} + \cdots + \frac{(n-1)^{\ell-h}}{n^{\ell-h+1}} + \left(\frac{n-1}{n} \right)^{\ell-h+1} \right) = \frac{1}{n^\ell},$$

whence, the claimed result follows. □

We can conclude that $M^{(\ell)}$ induces the following stochastic event:

$$p_{M^{(\ell)}}(\sigma^k) = |Q_{acc}^{(\ell)}| \cdot x_\ell(k) \begin{cases} = 0 & \text{if } k < \ell \\ \geq \left(\frac{n-1}{n}\right)^\ell & \text{if } k \geq \ell. \end{cases} \tag{5}$$

This shows that the automaton $M^{(\ell)}$ recognizes $\sigma^{\geq \ell}$ with isolated cut point and $n^{O(\ell)}$ basis states. As expected, for $n \to \infty$, the event in (5) approximates a deterministic behavior. In fact, for growing values of n, we have $p_{M^{(\ell)}}(\sigma^k) \to 1$ for $k \geq \ell$, and $p_{M^{(\ell)}}(\sigma^k) = 0$ for $k < \ell$.

To sum up, let us get back to our initial purpose, i.e., building an isolated cut point LQFA for the language $\sigma^{\geq T}$. Such a LQFA is obtained by pushing T steps ahead from $M^{(1)}$ the inductive construction to finally get the LQFA $M^{(T)}$. As noted, $M^{(T)}$ features $n^{O(T)}$ basis states, n being the number of basis states of $M^{(1)}$. From Equation (5), we can fix a cut point $\frac{1}{2} \cdot \left(\frac{n-1}{n}\right)^T$ isolated by $\frac{1}{2} \cdot \left(\frac{n-1}{n}\right)^T$. By increasing n, we widen such an isolation, tending to a deterministic recognition of the language $\sigma^{\geq T}$.

Focusing on the size of $M^{(T)}$, we observe that its number of basis states exponentially depends on T. As a matter of fact, we can avoid such an exponential blow up by noticing that even the LQFA $M^{(3)}$ can actually accept with isolated cut point the language $\sigma^{\geq T}$, for $T \geq 4$. This is due to the fact that the stochastic event induced by $M^{(3)}$ is an increasing function, as one may readily infer from Equation (3) and Figure 2. By this property, we can fix the isolated cut point between $p_{M^{(3)}}(\sigma^{T-1})$ and $p_{M^{(3)}}(\sigma^T)$, thus recognizing $\sigma^{\geq T}$ with $n^{O(1)}$ basis states, not depending on T any more. Nevertheless, such a dramatic size reduction comes at a price. In fact, the isolation around the cut point shrinks from $\frac{1}{2} \cdot \left(\frac{n-1}{n}\right)^T$ to $\frac{p_{M^{(3)}}(\sigma^T) - p_{M^{(3)}}(\sigma^{T-1})}{2} = \frac{1}{2} \cdot (\frac{2}{n})^{T-3} \cdot (\frac{n-1}{n})^{T-1} \cdot (\frac{n+1}{n})$. This isolation vanishes as n grows, thus suggesting to consider small values of n. E.g., for $n = 2$ we obtain an isolation of $\frac{3}{2} \cdot (\frac{1}{2})^T$; for $n = 3$ we get $\frac{27}{8} \cdot (\frac{4}{9})^T$.

4 Isolated Cut Point LQFAs for Unary Regular Languages

Here, we are going to use the LQFAs designed in the previous section as modules in a more general construction yielding isolated cut point LQFAs for unary regular languages. This investigation is inspired by [5] where the same problem is tackled for MM-QFAs. Our result constructively shows that isolated cut point MM-QFAs and LQFAs are equivalent on unary inputs, in sharp contrast to the case for general alphabets where MM-QFAs outperform LQFAs (see Section 2.3).

We start by observing that, according to Theorem 1, any unary regular language $L \subseteq \sigma^*$ can viewed as the disjoint union of two unary languages, namely, the finite language $L_T = L \cap \sigma^{\leq T}$ plus the ultimately periodic language $L_P = L \cap \sigma^{\geq T+1}$. So, we are going to design two LQFA modules recognizing these two languages with isolated cut point, and then suitably assemble such modules into a final isolated cut point LQFA A_L for the unary regular language L.

The finite language L_T: We define the "$(T+1)$-periodic continuation" $L_{T\circlearrowleft}$ of L_T, namely, the language obtained from L_T by adding all the strings of the form $\sigma^{i+h\cdot(T+1)}$, with $h \geq 0$, for $\sigma^i \in L_T$. Formally, $L_{T\circlearrowleft} = \left\{\sigma^{i+h\cdot(T+1)} \mid h \geq 0 \text{ and } \sigma^i \in L_T\right\}$. Clearly, $L_{T\circlearrowleft}$ is a periodic language of period $(T+1)$, and we have that $L_T = L_{T\circlearrowleft} \cap \sigma^{\leq T}$. Therefore, in order to recognize L_T, we start by defining the isolated cut point LQFA $A_{T\circlearrowleft}$ for $L_{T\circlearrowleft}$. We let $A_{T\circlearrowleft} = (Q, \{\sigma\}, \pi_0, \{U(\sigma), U(\sharp)\}, \{\mathscr{O}_\sigma, \mathscr{O}_\sharp\}, Q_{acc})$, where: $Q = \{q_0, \ldots, q_T\}$ is the set of basis states, $Q_{acc} = \left\{q_i \mid 0 \leq i \leq T \text{ and } \sigma^i \in L_T\right\}$ is the set of accepting basis states, $\pi_0 = \boldsymbol{e}_1$ is the initial superposition, $U(\sigma) = S$, where $S \in \{0,1\}^{(T+1)\times(T+1)}$ is the matrix representing the cyclic permutation: S has 1 at the $(i, i+1)$th entries for $1 \leq i \leq T$ and at the $(T+1, 1)$th entry, all the other entries are 0, $U(\sharp) = I^{(T+1)}$, $\mathscr{O}_\sigma$ is the observable having the identity as sole projector, $\mathscr{O}_\sharp$ is the usual final observable projecting onto the subspace spanned by Q_{acc}. Given the observable $\mathscr{O}_\sigma$, we have that $A_{T\circlearrowleft}$ is basically a MO-QFA whose induced event writes as $p_{A_{T\circlearrowleft}}(\sigma^k) = \|\pi_0 \cdot U(\sigma)^k \cdot U(\sharp) \cdot P_{acc}(\sharp)\|^2$. After processing the input $\sigma^k\sharp$, the state $\xi(k)$ of $A_{T\circlearrowleft}$ is

$$\xi(k) = \pi_0 \cdot U(\sigma)^k \cdot U(\sharp) = \boldsymbol{e}_1 \cdot U(\sigma)^k \cdot U(\sharp) = \boldsymbol{e}_{(k \bmod (T+1))+1}. \tag{6}$$

Let us now discuss measuring by the final observable, i.e., the action of the projector $P_{acc}(\sharp)$ on the final superposition $\xi(k)$. By (6), $\xi(k)$ is $\boldsymbol{e}_{(k \bmod (T+1))+1}$, representing the basis state $q_{k \bmod (T+1)}$. By definition of Q_{acc} we have that $q_{k \bmod (T+1)}$ is an accepting state if and only if $\sigma^{k \bmod (T+1)} \in L_T$ if and only if $\sigma^k \in L_{T\circlearrowleft}$. Therefore, we can rewrite the stochastic event induced by $A^{T^\circlearrowleft}$ as $p_{A_{T\circlearrowleft}}(\sigma^k) = \|\xi(k) \cdot P_{acc}(\sharp)\|^2 = 1$ if $\sigma^k \in L_{T\circlearrowleft}$, and 0 otherwise. Whence, the LQFA $A_{T\circlearrowleft}$ recognizes $L_{T\circlearrowleft}$ by a deterministic event. Now, we need $A_{T\circlearrowleft}$ to work simultaneously with a module which checks whether or not the input string has length not exceeding T, so that the resulting accepted language is $L_{T\circlearrowleft} \cap \sigma^{\leq T} = L_T$. Such a module can be obtained by complementing the LQFA $M^{(T+1)}$ for $\sigma^{\geq T+1}$ presented in Section 3 (basically, by taking $Q \setminus Q_{acc}$ as the set of accepting basis states). The resulting LQFA $\overline{M}^{(T+1)}$ induces the complement of the event in Equation (5) with $\ell = T+1$:

$$p_{\overline{M}^{(T+1)}}(\sigma^k) = 1 - p_{M^{(T+1)}}(\sigma^k) \begin{cases} = 1 & \text{if } k \leq T \\ \leq 1 - \left(\frac{n-1}{n}\right)^{(T+1)} & \text{if } k \geq T+1, \end{cases}$$

thus recognizing the language $\sigma^{\leq T}$ with isolated cut point and $n^{O(T)}$ basis states. Finally, we build the LQFA $A_{T\circlearrowleft} \otimes \overline{M}^{(T+1)}$ (basically by taking the direct product component wise of the two LQFAs $A_{T\circlearrowleft}$ and $\overline{M}^{(T+1)}$) inducing the product event

$$p_{A_{T\circlearrowleft} \otimes \overline{M}^{(T+1)}}(\sigma^k) = p_{A_{T\circlearrowleft}} \cdot p_{\overline{M}^{(T+1)}}(\sigma^k) \begin{cases} = 1 & \text{if } \sigma^k \in L_T \\ \leq 1 - \left(\frac{n-1}{n}\right)^{(T+1)} & \text{otherwise,} \end{cases}$$

defining L_T with $(T+1) \cdot n^{O(T)}$ basis states, and cut point $1 - \frac{1}{2} \cdot \left(\frac{n-1}{n}\right)^{(T+1)}$ isolated by $\frac{1}{2} \cdot \left(\frac{n-1}{n}\right)^{(T+1)}$. Notice that, for large values of n, the LQFA $A_{T\circlearrowleft} \otimes \overline{M}^{(T+1)}$ approximates a deterministic recognition of L_T.

The ultimately periodic language L_P: It suites our goal to rewrite L_P as $L_P = L_{P\circlearrowleft} \cap \sigma^{\geq T+1}$, where we let $L_{P\circlearrowleft} = \left\{\sigma^{(T+1+i) \bmod P + h\cdot P} \mid 0 \leq i < P,\ h \geq 0, \text{ and } \sigma^{T+1+i} \in L_P\right\}$. Clearly, $L_{P\circlearrowleft}$ is a periodic language of period P. So, for recognizing L_P, we first focus on building the isolated cut point LQFA $A_{P\circlearrowleft}$ for $L_{P\circlearrowleft}$. We let $A_{P\circlearrowleft} = (Q, \{\sigma\}, \pi_0, \{U(\sigma), U(\sharp)\}, \{\mathscr{O}_\sigma, \mathscr{O}_\sharp\}, Q_{acc})$, where: $Q = \{q_0, \ldots, q_{P-1}\}$ is the set of basis states, $Q_{acc} = \left\{q_i \mid 0 \leq i < P \text{ and } \sigma^i \in L_{P\circlearrowleft}\right\}$ is the set of accepting basis states, $\pi_0 = \boldsymbol{e}_1$ is the initial superposition, $U(\sigma) = S$, where $S \in \{0,1\}^{P\times P}$ is the cyclic permutation matrix, $U(\sharp) = I^{(P)}$, $\mathscr{O}_\sigma$

is the observable having the identity as sole projector, $\mathscr{O}_\sharp$ is the usual final observable projecting onto the subspace spanned by Q_{acc}. Given the observable $\mathscr{O}_\sigma$, we have that $A_{P^\circlearrowleft}$ is basically a MO-QFA whose induced event writes as $p_{A_{P^\circlearrowleft}}(\sigma^k) = \|\pi_0 \cdot U(\sigma)^k \cdot U(\sharp) \cdot P_{acc}(\sharp)\|^2$. After processing the input $\sigma^k\sharp$, the state $\xi(k)$ of $A_{P^\circlearrowleft}$ is

$$\xi(k) = \pi_0 \cdot U(\sigma)^k \cdot U(\sharp) = \boldsymbol{e}_1 \cdot U(\sigma)^k \cdot U(\sharp) = \boldsymbol{e}_{(k \bmod P)+1}. \tag{7}$$

Let us now measure the final observable on the final superposition $\xi(k)$. By (7), $\xi(k)$ is $\boldsymbol{e}_{(k \bmod P)+1}$, representing the basis state $q_{k \bmod P}$. By definition of Q_{acc}, we have that $q_{k \bmod P}$ is an accepting state if and only if $\sigma^k \in L_{P^\circlearrowleft}$. Therefore, the stochastic event induced by $A_{P^\circlearrowleft}$ is $p_{P^\circlearrowleft}(\sigma^k) = \|\xi(k) \cdot P_{acc}(\sharp)\|^2 = 1$, if $\sigma^k \in L_{P^\circlearrowleft}$, and 0 otherwise. whence, the LQFA $A_{P^\circlearrowleft}$ recognizes $L_{P^\circlearrowleft}$ by a deterministic event. Now, we need $A_{P^\circlearrowleft}$ to work simultaneously with a module which checks whether or not the input string has length exceeding T, so that the resulting accepted language is $L_{P^\circlearrowleft} \cap \sigma^{\geq T+1} = L_P$. Such a module is the LQFA $M^{(T+1)}$ for $\sigma^{\geq T+1}$ presented in Section 3, and inducing the event

$$p_{M^{(T+1)}}(\sigma^k) \begin{cases} \geq (\frac{n-1}{n})^{T+1} & \text{if } k \geq T+1 \\ = 0 & k \leq T, \end{cases}$$

thus recognizing the language $\sigma^{\geq T+1}$ with isolated cut point and $n^{O(T)}$ basis states. Finally, we build the LQFA $A_{P^\circlearrowleft} \otimes M^{(T+1)}$, inducing the product event

$$p_{A_{P^\circlearrowleft} \otimes M^{(T+1)}}(\sigma^k) = p_{A_{P^\circlearrowleft}} \cdot p_{M^{(T+1)}}(\sigma^k) \begin{cases} \geq (\frac{n-1}{n})^{T+1} & \text{if } \sigma^k \in L_P \\ = 0 & \text{otherwise,} \end{cases}$$

defining L_P with $P \cdot n^{O(T)}$ basis states, and cut point $\frac{1}{2} \cdot \left(\frac{n-1}{n}\right)^{(T+1)}$ isolated by $\frac{1}{2} \cdot \left(\frac{n-1}{n}\right)^{(T+1)}$. Notice that, for large values of n, the LQFA $A_{P^\circlearrowleft} \otimes M^{(T+1)}$ approximates a deterministic recognition of L_P.

Putting things together: We are now ready to suitably assemble the two LQFAs $A_T = A_{T^\circlearrowleft} \otimes \overline{M}^{(T+1)}$ and $A_P = A_{P^\circlearrowleft} \otimes M^{(T+1)}$ so far described to obtain an isolated cut point LQFA A_L for the unary regular language L. We notice that $L = L_T \cup L_P = (L_T^c \cap L_P^c)^c$. This suggests first to construct LQFAs for L_T^c and L_P^c by building $\overline{A}_T$ and $\overline{A}_P$ inducing the complement events $p_{\overline{A}_T} = 1 - p_{A_T}$ and $p_{\overline{A}_P} = 1 - p_{A_P}$, respectively. Next, to account for the intersection, we construct the LQFA $\overline{A}_L = \overline{A}_T \otimes \overline{A}_P$ inducing the product event $p_{\overline{A}_L} = (1 - p_{A_T}) \cdot (1 - p_{A_P})$. Finally, the desired LQFA A_L will be obtained by complementing $\overline{A}_L$, so that $p_{A_L} = (1 - p_{\overline{A}_L}) = 1 - (1 - p_{A_T}) \cdot (1 - p_{A_P}) = p_{A_T} + p_{A_P} - p_{A_T} \cdot p_{A_P}$.

Let us now explain how p_{A_L} behaves on input string σ^k:

- $\sigma^k \in L = L_T \cup L_P$: Clearly, we have either $\sigma^k \in L_T$ or $\sigma^k \in L_P$. Suppose $\sigma^k \in L_T$. Then, we have that $p_{A_T}(\sigma^k) = 1$ since $A_T = A_{T^\circlearrowleft} \otimes \overline{M}^{(T+1)}$ and both its sub-modules will accept with certainty; correspondingly, $p_{A_P}(\sigma^k) = 0$ since $A_P = A_{P^\circlearrowleft} \otimes M^{(T+1)}$ and the sub-module $M^{(T+1)}$ accepts with 0 probability the input strings of length less than or equal to T. Globally, we have $p_{A_L}(\sigma^k) = 1$. Suppose $\sigma^k \in L_P$. Then, we have that $p_{A_P}(\sigma^k) \geq \left(\frac{n-1}{n}\right)^{T+1}$ since $A_{P^\circlearrowleft}$ accepts with certainty, while the sub-module $M^{(T+1)}$ accepts with probability not less than $\left(\frac{n-1}{n}\right)^{T+1}$. Let us now focus on A_T. The sub-module $A_{T^\circlearrowleft}$ could accept with probability either 0 or 1. In the former case, globally we have $p_{A_L}(\sigma^k) \geq \left(\frac{n-1}{n}\right)^{T+1}$, in the latter, the sub-module $\overline{M}^{(T+1)}$ accepts with a probability bounded above by $1 - \left(\frac{n-1}{n}\right)^{T+1}$. By letting $(1-y)$ the acceptance probability of $\overline{M}^{(T+1)}$, with $0 \leq y \leq \left(\frac{n-1}{n}\right)^{T+1}$, we get $p_{A_L}(\sigma^k) \geq \left(\frac{n-1}{n}\right)^{T+1} + (1-y) - \left(\frac{n-1}{n}\right)^{T+1} \cdot (1-y) \geq \left(\frac{n-1}{n}\right)^{T+1}$.

  In conclusion, for any $\sigma^k \in L$, we have $p_{A_L}(\sigma^k) \geq \left(\frac{n-1}{n}\right)^{T+1}$.

- $\sigma^k \notin L = L_T \cup L_P$: Clearly, both $\sigma^k \notin L_T$ and $\sigma^k \notin L_P$. By assuming $k \leq T$, we must have $\sigma^k \notin L_{T^{\circlearrowleft}}$. Therfore, the sole acceptance probability contribution could come from the module $A_P = A_{P^{\circlearrowleft}} \otimes M^{(T+1)}$. However, since $k \leq T$, the sub-module $M^{(T+1)}$ accepts with 0 probability. So, $p_{A_L}(\sigma^k) = 0$. Instead, by assuming $k \geq T+1$, we must have that $\sigma^k \notin L_{P^{\circlearrowleft}}$. Thus, the sole acceptance probability could come from the module A_T. However, the acceptance probability yielded by the sub-module $\overline{M}^{(T+1)}$ turns out to be at most $1 - \left(\frac{n-1}{n}\right)^{T+1}$.

  In conclusion, for any $\sigma^k \notin L$, we have $p_{A_L}(\sigma^k) \leq 1 - \left(\frac{n-1}{n}\right)^{T+1}$.

Summing up, the stochastic event induced by the LQFA A_L is

$$p_{A_L}(\sigma^k) \begin{cases} \geq \left(\frac{n-1}{n}\right)^{T+1} & \text{if } \sigma^k \in L \\ \leq 1 - \left(\frac{n-1}{n}\right)^{T+1} & \text{otherwise.} \end{cases} \tag{8}$$

By the event in Equation (8), we get that A_L recognizes L with the following cut point and isolation radius:

$$\lambda = \frac{1}{2} \cdot \left(\left(\frac{n-1}{n}\right)^{T+1} + 1 - \left(\frac{n-1}{n}\right)^{T+1} \right) = \frac{1}{2}, \quad \rho = \frac{1}{2} \cdot \left(\left(\frac{n-1}{n}\right)^{T+1} - 1 + \left(\frac{n-1}{n}\right)^{T+1} \right) = \left(\frac{n-1}{n}\right)^{T+1} - \frac{1}{2}.$$

Clearly, to have an isolation around λ, we must require that $\rho > 0$. This can always be achieved on any $T > 0$ by imposing $\left(\frac{n-1}{n}\right)^{T+1} > \frac{1}{2}$, which is attained whenever $n > \frac{1}{1 - \sqrt[T+1]{\frac{1}{2}}}$. This latter condition is satisfied, e.g., by letting $n = 4T$ for any $T > 0$. Nevertheless, the isolation radius ρ tends to $\frac{1}{2}$ as n grows.

Let us inspect the size of the LQFA $A_L = \overline{\overline{A}_T \otimes \overline{A}_P}$. As above pointed out, A_T and A_P have, respectively, $(T+1) \cdot n^{O(T)}$ and $P \cdot n^{O(T)}$ basis states. The complements $\overline{A}_T$ and $\overline{A}_P$ maintain the same number of basis states, while the product $\overline{A}_T \otimes \overline{A}_P$ requires $((T+1) \cdot n^{O(T)}) \cdot (P \cdot n^{O(T)}) \leq T \cdot P \cdot n^{O(T)}$ basis states. The final complement $\overline{\overline{A}_T \otimes \overline{A}_P}$ maintains the same number of basis states. By replacing n with $4T$, as above suggested, the number of basis states of the isolated cut point LQFA A_L for L becomes $P \cdot T^{O(T)}$.

5 Conclusions

In this work, we have exhibited a modular framework for building isolated cut point LQFAs for unary regular languages. By suitably adapting to the unary case an inductive construction in [1, 14], we have first designed LQFAs discriminating unary inputs on the basis of their length. These devices have then been plugged into two sub-modules recognizing the finite part and the ultimately periodic part any unary regular language consists of. The resulting LQFA recognizes a unary regular language L with isolated cut point $\frac{1}{2}$, and a number of basis states which is exponential in the number of states of the minimal DFA for L. In spite of this exponential size blow up, it should be stressed that more restricted models of quantum finite automata in the literature, such as MO-QFAs, cannot recognize all unary regular languages. On the other hand, a linear amount of basis states is sufficient for the more powerful model of isolated cut point MM-QFAs [5]. Thus, it would be worth investigating whether a more size efficient construction for unary LQFAs could be provided. Another interesting line of research might explore the descriptional power (see, e.g., [2, 8, 12, 13] for topics in descriptional complexity) of isolated cut point LQFAs with respect to other relevant classes of subregular languages such as, e.g., commutative regular languages [20].

Acknowledgements. The authors wish to thank the anonymous referees for their valuable comments.

References

[1] Andris Ambainis, Martin Beaudry, Marats Golovkins, Arnolds Kikusts, Mark Mercer & Denis Thérien (2006): *Algebraic results on quantum automata*. Theory of Computing Systems 39, pp. 165–188, doi:10.1007/s00224-005-1263-x.

[2] Zuzana Bednárová, Viliam Geffert, Carlo Mereghetti & Beatrice Palano (2017): *Boolean language operations on nondeterministic automata with a pushdown of constant height*. Journal of Computer and System Sciences 90, pp. 99 – 114, doi:10.1016/j.jcss.2017.06.007.

[3] Maria Paola Bianchi, Carlo Mereghetti & Beatrice Palano (2014): *Size lower bounds for quantum automata*. Theoretical Computer Science 551, p. 102 – 115, doi:10.1016/j.tcs.2014.07.004.

[4] Maria Paola Bianchi, Carlo Mereghetti & Beatrice Palano (2017): *Quantum finite automata: Advances on Bertoni's ideas*. Theoretical Computer Science 664, pp. 39–53, doi:10.1016/j.tcs.2016.01.045.

[5] Maria Paola Bianchi & Beatrice Palano (2010): *Behaviours of unary quantum automata*. Fundamenta Informaticae 104, pp. 1–15, doi:10.3233/FI-2010-333.

[6] Alex Brodsky & Nicholas Pippenger (2002): *Characterizations of 1-way quantum finite automata*. SIAM Journal on Computing 31, pp. 1456–1478, doi:10.1137/S0097539799353443.

[7] Alessandro Candeloro, Carlo Mereghetti, Beatrice Palano, Simone Cialdi, Matteo G. A. Paris & Stefano Olivares (2021): *An enhanced photonic quantum finite automaton*. Applied Sciences 11, p. 8768, doi:10.3390/app11188768.

[8] Markus Holzer & Martin Kutrib (2011): *Descriptional and computational complexity of finite automata - A survey*. Information and Computation 209, pp. 456–470, doi:10.1016/j.ic.2010.11.013.

[9] John E. Hopcroft, Rajeev Motwani & Jeffrey D. Ullman (2006): *Introduction to Automata Theory, Languages, and Computation*, 3rd edition. Addison-Wesley.

[10] R.I.G. Hughes (1992): *The Structure and Interpretation of Quantum Mechanics*. Harvard University Press.

[11] Attila Kondacs & John Watrous (1997): *On the power of quantum finite state automata*. In: Proc. 38th Symp. on Found. Comp. Sci. (FOCS), IEEE Computer Society, pp. 66–75, doi:10.1109/SFCS.1997.646094.

[12] Martin Kutrib, Andreas Malcher, Carlo Mereghetti & Beatrice Palano (2020): *Deterministic and nondeterministic iterated uniform finite-state transducers: computational and descriptional power*. In: Proc. 16th Int. Conf. Comp. Europe (CiE 2020), LNCS 12098, Springer, pp. 87–99, doi:10.1007/978-3-030-51466-2_8.

[13] Martin Kutrib, Andreas Malcher, Carlo Mereghetti & Beatrice Palano (2020): *Iterated uniform finite-state transducers: descriptional complexity of nondeterminism and two-way motion*. In: Proc. 22nd Int. Conf. Des. Comp. Form. Sys. (DCFS 2020), LNCS 12442, Springer, pp. 117–129, doi:10.1007/978-3-030-62536-8_10.

[14] Mark Mercer (2007): *Applications of Algebraic Automata Theory to Quantum Finite Automata*. Ph.D. thesis, McGill University, Montreal, Quebec, Canada.

[15] Carlo Mereghetti, Beatrice Palano, Simone Cialdi, Valeria Vento, Matteo G. A. Paris & Stefano Olivares (2020): *Photonic realization of a quantum finite automaton*. Physical Review Research 2, p. 013089, doi:10.1103/PhysRevResearch.2.013089.

[16] Cristopher Moore & James P. Crutchfield (2000): *Quantum automata and quantum grammars*. Theoretical Computer Science 237, pp. 275–306, doi:10.1016/S0304-3975(98)00191-1.

[17] Rohit Parikh (1966): *On context-free languages*. J. ACM 13, pp. 570–581, doi:10.1145/321356.321364.

[18] Michael O. Rabin (1963): *Probabilistic automata*. Information and Control 6, pp. 230–245, doi:10.1016/S0019-9958(63)90290-0.

[19] Georgy E. Shilov (1971): *Linear Algebra*. Prentice-Hall. Reprinted by Dover, 1977.

[20] Bianca Truthe (2018): *Hierarchy of Subregular Language Families*. Technical Report, Universitätsbibliothek Gießen, Institut für Informatik, doi:10.22029/JLUPUB-6984.