On the computational power of affine automataa

Mika Hirvensalo[1] ⋆, Etienne Moutot[2]⋆⋆, and Abuzer Yakaryılmaz[3]⋆⋆⋆

[1] Department of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland mikhirve@utu.fi

[2] Département d'Informatique, École Normale Supérieure de Lyon, 69007 Lyon, France etienne.moutot@ens-lyon.org

[3] Faculty of Computing, University of Latvia, Raina bulv. 19, Riga, LV-1586, Latvia abuzer@lu.lv

Abstract. We investigate the computational power of affine automata (AfAs) introduced in [5]. We first present a simplified proof for changing the cutpoint and a method for reducing the error rate. We then address to the question of [5] by showing that any affine language can be recognized by an AfA with certain limitation on the entries of affine states and transition matrices. Finally, we present the first examples of languages that cannot be recognized by AfAs with bounded-error.

Keywords: affine automata, cutpoint languages, bounded error, compact sets, error reduction

1 Introduction

Finite automata are interesting computational models because of their simplicity, compared to more complex models like pushdown automata or Turing machines. They also represent a very concrete restriction on computation: they only have a finite memory. A lot of different automata models have been studied during the years, such as deterministic [11], probabilistic [9] and quantum [3] ones. All these models share two a common features: The state vector set is compact and the acceptance function mapping the final state vector into real interval $[0, 1]$ can be interpreted linear. The linearity is desirable because of mathematical simplicity, but on the other hand, it may represent a limitation on the computational power.

Recently, A. Díaz-Caro and A. Yakaryılmaz introduced a new model, called *affine computation* [5], also investigated in [14] and [4]. As a non-physical model, the goal of affine computation is to investigate on the power of interference caused by negatives amplitudes in the computation, like in the quantum case. But unlike quantum automata, affine ones have unbounded state set and the final operation corresponding to quantum measurement cannot be interpreted as linear. The final operation in affine automata is analogous to renormalization in Kondacs-Watrous [7] and Latvian [2] quantum automata models.

⋆ Partially supported by Väisala Foundation

⋆⋆ Partially supported by TUCS COM^3-project

⋆⋆⋆ Partially supported by TUCS COM^3-project and ERC Advanced Grant MQC

In this paper, we present some stability results (Section 3): We use classical constructions of tensoring and direct sum *simultaneously* to get a simpler proof for the fact that the cutpoint of affine languages can be changed. The aforementioned operations are also used to produce an error reduction method in bounded error case. It should be emphasized here that in the case of (one-way) probabilistic and quantum automata, the constraint of bounded error implies the regularity of accepted language, and hence there is always a zero-error (deterministic) automaton accepting the same language. Therefore, the error reduction technique is apparently uninteresting when restricting to automata with compact state set.

Any entry of an affine state vector or a transition matrix can be arbitrarily away from zero. However, here we show that (Section 4) any affine language can be recognized by an AfA providing that any such entry can be in the interval $[-1, 1]$. This partially answers to the question implicitly proposed in [5]: In the case of unbounded error, affine automata computations can be performed with a compact state vector set, and hence in that case, the power of affine automata seems to originate rather from the nonlinear nature of the final value function.

Finally, we present (Section 5) the first languages shown not to be recognized by any bounded-error AfA and conclude in Section 6.

2 Preliminaries

We use notation Σ for the input alphabet and ε for the empty string .

Probabilistic automata are a generalization of deterministic finite automata that can make random choices [10].

Formally, a *probabilistic finite automaton* (PFA) P is a 5-tuple

$$P = (E, \Sigma, \{M_x \mid x \in \Sigma\}, \boldsymbol{v}_0, E_a)$$

where $E = \{e_1, \ldots, e_k\}$ is the finite set of states of P, $\{M_x \mid x \in \Sigma\}$ is the set of stochastic transition matrices, $\boldsymbol{v}_0$ is the initial probabilistic state (the probability distribution on the states), and $E_a \subseteq E$ is the set of accepting states.

The computation starts in $\boldsymbol{v}_0$, and then the given input, say $w = w_1 \cdots w_n \in \Sigma^*$ for some $n > 0$, is read once from left to right symbol by symbol and for each symbol the corresponding transition matrix is applied:

$$\boldsymbol{v}_f = M_w \boldsymbol{v}_0 = M_{w_n} \ldots M_{w_1} \boldsymbol{v}_0.$$

Remark that if $w = \varepsilon$, $\boldsymbol{v}_f = \boldsymbol{v}_0$. The *accepting probability* of P on w is given by

$$f_P(w) = \boldsymbol{p} M_w \boldsymbol{v}_0, \tag{1}$$

where $\boldsymbol{p} = \begin{pmatrix} \delta_1 & \ldots & \delta_k \end{pmatrix}$ and $\delta_i = 0$ if and only if $e_i \in E_a$.

Affine automata are a generalization of PFAs allowing negative transition values. Only allowing negative values in the transition matrices does not add any power (generalized probabilistic automata are equivalent to usual ones [12]), but affine automata introduces also a non-linear behaviour. The automaton acts like usual generalized probabilistic automaton until the last operation, a non-linear operation called *weighting*.

A vector $\boldsymbol{v} \in \mathbb{R}^n$ is *an affine vector* if and only if its coordinates sums up to 1. A matrix M is *an affine matrix* if and only if all its columns are affine vectors. Note that if M and N are affine matrices , then MN is also an affine matrix. In particular, if $\boldsymbol{v}$ is an affine vector, then $M\boldsymbol{v}$ is also an affine vector.

Formally, an *affine finite automaton* (AfA) A is a 5-tuple

$$A = (E, \Sigma, \{M_x \mid x \in \Sigma\}, \boldsymbol{v}_0, E_a)$$

where all components exactly the same as for probabilistic automata by replacing stochastic property with affine one in the initial state and transition matrices.

As in PFAs, after reading a word $w = w_1 \dots w_n$, the final state of A is $\boldsymbol{v}_f = M_w \boldsymbol{v}_0$ like in the probabilistic case, but the function $f_A : \Sigma^* \to [0,1]$ computed by A is defined as

$$f_A(w) = \frac{\sum_{e_i \in E_a} |(v_f)_i|}{\sum_{e_i \in E} |(v_f)_i|}, \tag{2}$$

and referred as the *accepting value* of A on w. Similar to projective measurements, we can rewrite Eq. 2 as given below. First, we define a projection matrix based on E_a: $P_A = P = \begin{pmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_n \end{pmatrix}$, where $\delta_i = \begin{cases} 1 \text{ if } e_i \in E_a \\ 0 \text{ otherwise} \end{cases}$.

Then, we can denote $f_A(\cdot)$ as

$$f_A(w) = \frac{|PM_w \boldsymbol{v}_0|}{|M_w \boldsymbol{v}_0|}. \tag{3}$$

Notice that the final value for PFA P (1) is defined as matrix product $\boldsymbol{v}_f \mapsto \boldsymbol{p}.\boldsymbol{v}_f$, which is a linear operation on $\boldsymbol{v}_f$. On the other hand, computing final value from $\boldsymbol{v}_f$ as in (3) involves nonlinear operations $\boldsymbol{v}_f \mapsto \dfrac{|P\boldsymbol{v}_f|}{|\boldsymbol{v}_f|}$. The renormalization here is analogous to those in Kondacs-Watrous [7] and Latvian [2] quantum automata models.

Given a function $f : \Sigma^* \to [0,1]$ computed by an automaton (stochastic or affine), there are different ways of defining the language of a PFA. The natural one is as follows: A language $L \subseteq \Sigma^*$ is recognized by an automaton A with cutpoint λ if and only if

$$L = \{w \in \Sigma^* \mid f_A(w) > \lambda\}.$$

These languages are called cutpoint languages. In the case of probabilistic (resp. affine automata), the set of cut-point languages are called *stochastic languages* (resp. *affine languages*) and denoted by SL (resp. AfL).

A stronger condition is to impose that accepted and rejected words are separated by a gap: the cutpoint is said to be isolated: A language L is recognized by an automaton A with *isolated cutpoint* λ if and only if there exist $\delta > 0$ such that $\forall w \in L, f_A(w) \geq \lambda + \delta$, and $\forall w \notin L, f_A(w) \leq \lambda - \delta$.

As we shall see, for affine automata it is always possible to shift the cutpoint $\lambda \in (0,1)$ to $\lambda = \frac{1}{2}$, and hence this notion of isolated cutpoint becomes equivalent to the bounded error recognition: Language $L \subseteq \Sigma^*$ is said to be recognized by an automaton A with *bounded error* if and only if there exists $\varepsilon > 0$ such that $\forall w \in L, f_A(w) \geq 1 - \varepsilon$, and $\forall w \notin L, f_A(w) \leq \varepsilon$.

The set of languages recognized with *bounded error* (or isolated cutpoint) affine automata is denoted by BAfL.

A classical result by Rabin [10] shows that isolated cutpoint stochastic languages are regular (denoted REG). Rabin's proof essentially relies on two facts: 1) the function mapping the final vector into $[0,1]$ is a contraction, and 2) the state vector set is bounded.

By modifying Rabin's proof, it is possible to show that also many quantum variants of stochastic automata obey the same principle: bounded-error property implies the regularity of the accepted languages. In fact, E. Jeandel generalized Rabin's proof by demonstrating that the compactness of the state vector set together with the continuity of the final function are sufficient to guarantee the regularity of the accepted language, if the cutpoint is isolated [6].

In the affine case however, the vector states do not lie in a compact set, we cannot prove that $\mathsf{BAfL} = \mathsf{REG}$ like in the probabilistic (or even quantum) case [6]. In fact, it is even the contrary: $\mathsf{REG} \subsetneq \mathsf{BAfL}$ [5].

We close this section by three basic facts. The following three operations on the state sets will be useful, when constructing new (affine) automata from the existing ones:
- $\overline{E} = \{e_i \mid e_i \notin E\}$ the complement of E,
- $E_a \times E_b = \{(e_i, e_j) \mid e_i \in E_a, e_j \in E_b\}$ the Cartesian product of E_a and E_b,
- $E_a \cup E_b = \{e_i \mid e_i \in E_a \text{ or } e_i \in E_b\}$ the union of E_a and E_b.

The following lemmata show how to formulate the above operations by using the formalism of projection matrices. The proofs are simple and we omit them.

Lemma 1. *Let E be the set of all states, $E_a, E_b \subseteq E$ and P_a, P_b the projections associated to them. Then*
- *P is the projection associated to the complement $\overline{E_a}$ if and only if $P = I - P_a$,*
- *P is the projection associated to $E_a \times E_b$ if and only if $P = P_a \otimes P_b$.*

Lemma 2. *Let E be the set of all states, $E_a, E_b \subseteq E$, such that $E_a \cap E_b = \emptyset$. Let P_a and P_b the projections associated to them. Then,*
- *P is the projection associated to $E_a \cup E_b$ if and only if $P = P_a + P_b$,*
- *For any matrix M and vector $\boldsymbol{v}$, $|PM\boldsymbol{v}| = |P_aM\boldsymbol{v}| + |P_bM\boldsymbol{v}|$.*

Lemma 3. *If A and B are affine matrices, then $A \otimes B$ is also affine. Moreover, $|A \otimes B| = |A||B|$.*

3 Stability Results

The main results of this section are stability results. The first are about the function of affine automata. They provide a way to prove an error reduction theorem. We then use this theorem to show the stability of bounded-error affine languages under intersection and union.

Proposition 1. *Let f, g be functions computed by affine automata, then there exists an affine automaton $\mathcal{C}$ such that $f_{\mathcal{C}} = fg$.*

Proof. The proof is the same as the stochastic case and essentially relies on the property of tensor product of Lemma 3. □

It is easy to design a 2-state PFA P such that $f_P : \Sigma^* \to \alpha$ for $\alpha \in [0, 1]$. Thus:

Corollary 1 *Let f be a function computed by an AfA and $\alpha \in [0, 1]$, then there exists an AfA $\mathcal{C}$ such that $f_{\mathcal{C}} = \alpha f$.*

Proposition 2. *Let f, g be functions computed by some AfAs and $\alpha, \beta \geq 0$ such that $\alpha + \beta = 1$, then there exists an AfA $\mathcal{C}$ such that $f_{\mathcal{C}} = \alpha f + \beta g$.*

Proof. Let $\mathcal{A} = (E^A, \Sigma, \{A_x\}, \boldsymbol{v}_0^A, E_a^A)$ and $\mathcal{B} = (E^B, \Sigma, \{B_x\}, \boldsymbol{v}_0^B, E_a^B)$ two automata such that $f = f_{\mathcal{A}}$ and $g = f_{\mathcal{B}}$. The idea here is to make two copies of $\mathcal{A} \otimes \mathcal{B}$ working in parallel, one having the final states of $\mathcal{A}$, the other the final states of $\mathcal{B}$. We define $\mathcal{C} = (E^C, \Sigma, \{C_x\}, \boldsymbol{v}_0^C, E_a^C)$ by:

$$C_x = \left(\begin{array}{c|c} A_x \otimes B_x & 0 \\ \hline 0 & A_x \otimes B_x \end{array}\right), \boldsymbol{v}_0^C = \left(\begin{array}{c} \alpha(\boldsymbol{v}_0^A \otimes \boldsymbol{v}_0^B) \\ \hline \beta(\boldsymbol{v}_0^A \otimes \boldsymbol{v}_0^B) \end{array}\right), P^C = \left(\begin{array}{c|c} P^A \otimes I_n & 0 \\ \hline 0 & I_k \otimes P^B \end{array}\right),$$

with P^A, P^B and P^C be the projections on E_a^A, E_a^B and E_a^C. Thus,

$$\begin{aligned} f_{\mathcal{C}}(w) &= \frac{\alpha|(P^A \otimes I_n)(A_x \otimes B_x)(\boldsymbol{v}_0^A \otimes \boldsymbol{v}_0^B)| + \beta|(I_k \otimes P^B)(A_x \otimes B_x)(\boldsymbol{v}_0^A \otimes \boldsymbol{v}_0^B)|}{(\alpha + \beta)|(A_x \otimes B_x)(\boldsymbol{v}_0^A \otimes \boldsymbol{v}_0^B)|} \\ &= \alpha \frac{|P^A A_w \boldsymbol{v}_0^A|}{|A_w \boldsymbol{v}_0^A|} + \beta \frac{|P^B B_w \boldsymbol{v}_0^B|}{|B_w \boldsymbol{v}_0^B|} = \alpha f(w) + \beta g(w). \end{aligned}$$ □

The first consequence of these stability results is a really short proof for shifting the cutpoint of an affine automaton. Although the construction in [5] gives a much more compact automata in term of number of states, our construction is simpler, and does not require as many specific cases.

Proposition 3. *Let $\mathcal{A}$ be and affine automaton and $\lambda_1, \lambda_2 \in [0, 1]$. There exists an affine automaton $\mathcal{B}$ such that*
- *$f_{\mathcal{A}}(w) > \lambda_1 \Leftrightarrow f_{\mathcal{B}}(w) > \lambda_2$ and*
- *$f_{\mathcal{A}}(w) = \lambda_1 \Leftrightarrow f_{\mathcal{B}}(w) = \lambda_2$.*

Proof. First we suppose $\lambda_1 \neq 1$. Let $\mathcal{B}$ the automaton such that $f_{\mathcal{B}} = \alpha f_{\mathcal{A}} + (1 - \alpha)1$, with $\alpha = \frac{1-\lambda_2}{1-\lambda_1}$. Then $f_{\mathcal{A}} > \lambda_1 \Rightarrow f_{\mathcal{B}} > \frac{(1-\lambda_2)\lambda_1 + \lambda_2 - \lambda_1}{1-\lambda_1} = \lambda_2$. And one has the same with $=$ or $<$.

For $\lambda_1 = 1$ it is even simpler, one has just to "resize" the function by taking $\mathcal{B}$ such that $f_{\mathcal{B}} = \lambda_2 f_{\mathcal{A}}$. And then, $f_{\mathcal{A}} = 1 \Rightarrow f_{\mathcal{B}} = \lambda_2$, and same for $<$. □

Using the same kind of construction we can prove that bounded-error mode, it is always possible to reduce the error. Reducing the error means increasing the gap between accepted and rejected words. The error probability could even be made as close to zero as one wants.

Lemma 4. *Let f be a function computed by affine automaton, then there exists an affine automaton $\mathcal{B}$ such that $f_{\mathcal{B}} = f^2(3-2f)$.*

Proof. Let $\mathcal{A} = (E, \Sigma, \{A_x\}, \boldsymbol{v}_0, E_a)$ such that $f = f_{\mathcal{A}}$. The automaton $\mathcal{B}$ will run 3 copies of $\mathcal{A}$ in parallel, and its final states are made to accept if 2 or 3 copies of $\mathcal{A}$ accept and reject otherwise (i.e. taking the majority answer). Formally, $\mathcal{B} = (E \otimes E \otimes E, \Sigma, \{B_x\}, \boldsymbol{v}_0', E_a')$ with

$$B_x = A_x \otimes A_x \otimes A_x,$$

$$\boldsymbol{v}_0' = \boldsymbol{v}_0 \otimes \boldsymbol{v}_0 \otimes \boldsymbol{v}_0,$$

$$E_a' = (E_a \times E_a \times E_a) \cup \left(\overline{E_a} \times E_a \times E_a\right) \cup \left(E_a \times \overline{E_a} \times E_a\right) \cup \left(E_a \times E_a \times \overline{E_a}\right).$$

Note that the four sets in parenthesis are all pairwise disjoints. Let P and P' be the projections associated to E_a and E_a'. Then,

$$P' = P \otimes P \otimes P + (I-P) \otimes P \otimes P + P \otimes (I-P) \otimes P + P \otimes P \otimes (I-P).$$

And by Lemma 1,

$$\begin{aligned} f_{\mathcal{B}}(w) = \frac{|P' B_w \boldsymbol{v}_0'|}{|B_w \boldsymbol{v}_0'|} &= \frac{|PA_w\boldsymbol{v}_0|^3 + 3|PA_w\boldsymbol{v}_0|\left(|A_w\boldsymbol{v}_0| - |PA_w\boldsymbol{v}_0|\right)}{|A_w\boldsymbol{v}_0|^3} \\ &= f(w)^3 + 3f(w)^2(1-f(w)) \\ &= f(w)^2(3-2f(w)). \end{aligned}$$

□

Proposition 4 (Error reduction). *Let $L \in \mathsf{BAfL}$. There exists an affine automaton $\mathcal{A}$ such that:*
- $\forall w \in L, f_{\mathcal{A}}(w) \geq \frac{3}{4}$
- $\forall w \notin L, f_{\mathcal{A}}(w) \leq \frac{1}{4}$

Proof. We do not give the details here, but the idea is as follows: Mapping $x \to x^2(3-2x)$ has attracting points at $x = 0$ and $x = 1$. Iterating the mapping, any point $x \in [0,1] \setminus \{\frac{1}{2}\}$ will tend to 0 (if $x < \frac{1}{2}$) or to 1 (if $x > \frac{1}{2}$). □

This technique could be applied to get any constant instead of $\frac{1}{4}$, to have an error bound as small as one wants.

This error reduction theorem also applies to probabilistic automata, but is not very interesting because in the probabilistic case it is known that bounded-error languages are exactly regular languages [6], and hence the error probability could always be 0. In our case, bounded-error languages are more complex than regular languages. But thanks to this error reduction they are stable under union, intersection and complement, just like regular languages.

Proposition 5. *Let* $L_A, L_B \in \mathsf{BAfL}$. *Then*
- $L_A \cup L_B \in \mathsf{BAfL}$,
- $L_A \cap L_B \in \mathsf{BAfL}$,
- $\overline{L_A} \in \mathsf{BAfL}$.

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be automata recognizing L_A and L_B with error bound ε at most $\frac{1}{4}$ (thanks to Proposition 4). We define $\mathcal{C}$ and $\mathcal{D}$ such that $f_{\mathcal{C}} = \frac{1}{2}(f_{\mathcal{A}} + f_{\mathcal{B}})$ and $f_{\mathcal{D}} = f_{\mathcal{A}} f_{\mathcal{B}}$. Let $w \in \Sigma^*$. We study the 4 possible options depending on the membership of w to L_A and L_B.
- $w \in L_A, w \in L_B$ (i.e. $w \in L_A \cup L_B, w \in L_A \cap L_B$) $\Rightarrow f_C \geq \frac{3}{4}$ and $f_{\mathcal{D}} \geq \frac{9}{16}$,
- $w \in L_A, w \notin L_B$ (i.e. $w \in L_A \cup L_B, w \notin L_A \cap L_B$) $\Rightarrow f_C \geq \frac{3}{8}$ and $f_{\mathcal{D}} \leq \frac{1}{4}$,
- $w \notin L_A, w \in L_B$ (i.e. $w \in L_A \cup L_B, w \notin L_A \cap L_B$) $\Rightarrow f_C \geq \frac{3}{8}$ and $f_{\mathcal{D}} \leq \frac{1}{4}$,
- $w \notin L_A, w \notin L_B$ (i.e. $w \notin L_A \cup L_B, w \notin L_A \cap L_B$) $\Rightarrow f_C \leq \frac{1}{4}$ and $f_{\mathcal{D}} \leq \frac{1}{16}$.

Because $\frac{3}{8} > \frac{1}{4}$ and $\frac{9}{16} > \frac{1}{4}$, $\mathcal{C}$ and $\mathcal{D}$ are deciding $L_A \cup L_B$ and $L_A \cap L_B$ with bounded error.

For the complement one has just to make a copy of $\mathcal{A}$ with accepting states $\overline{E_a}$. The resulting function will be $1 - f_{\mathcal{A}}$, leading to accept the rejected words of $\mathcal{A}$ and vice-versa. □

4 Equivalent Forms of Affine Automata

General affine automata are hard to study because of the lack of structure of their transition matrices and state vectors. We provide here some equivalent forms which have more restrictive properties. These equivalent forms are useful not only because it provides simpler equivalent models but also because they provide a way understand the power of affine computation.

The first result is that assuming the initial affine (probabilistic) state as the first deterministic state does not change the power of AfAs (PFAs); the proof is the same as for the probabilistic automata.

Proposition 6. *Let* $\mathcal{A}$ *be an affine automaton with* n *states, there exist* $\mathcal{B}$ *with* $n+1$ *states with the initial state* $(1, 0, \ldots, 0)$ *and such that* $f_{\mathcal{A}} = f_{\mathcal{B}}$.

Proof. Let $\mathcal{A} = (E, \Sigma, \{A_x\}, \boldsymbol{v}_0, E_a)$. Then, $\mathcal{B} = (E \cup \{e'\}, \Sigma, \{B_x\}, \boldsymbol{v}'_0, E_a)$, with $\boldsymbol{v}'_0 = (1, 0, \ldots, 0)^T$ and $B_x = \left(\begin{array}{c|c} 0 & 0 \ldots 0 \\ \hline A_x \boldsymbol{v}_0 & A_x \end{array}\right)$. Thus we can follow $f_{\mathcal{B}} = f_{\mathcal{A}}$ by

$$B_w \boldsymbol{v}'_0 = B_{w_n} \ldots B_{w_2} B_{w_1} \boldsymbol{v}'_0 = \left(\begin{array}{c|c} 0 & 0 \ldots 0 \\ \hline A_w \boldsymbol{v}_0 & A_w \end{array}\right) \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ A_w \boldsymbol{v}_0 \end{pmatrix}.$$ □

Then we prove that one could also assume that all state vectors and transition matrices have coefficients only in $[-1, 1]$.

Proposition 7. *Any language in* AfL *can be recognized by a AfA $\mathcal{B}$ with cutpoint $\frac{1}{2}$ such that each entry of affine states during the computation is always in $[-1,1]$.*

Proof. Let $\mathcal{A} = (E = \{e_1, \ldots, e_k\}, \Sigma, \{A_x\}, v_0 = (1, 0, \ldots, 0)^T, E_a)$ be an AfA such that $w \in L \Leftrightarrow f_{\mathcal{A}}(w) > \frac{1}{2}$, and $C = \max_{x,i,j} |(A_x)_{i,j}|$. Then, $\mathcal{B}$ is as follows:

$$\mathcal{B} = (E \cup \{e_{n+1}, e_{n+2}\}, \Sigma, \{B_x\}, \boldsymbol{v}'_0, E_a \cup \{e_{n+1}\}) \quad \text{with}$$

$$B_x = \frac{1}{2kC} \left(\begin{array}{ccc|cc} & & & 0 & 0 \\ & 2A_x & & \vdots & \vdots \\ & & & 0 & 0 \\ \hline kC-1 & \ldots & kC-1 & 2kC & 0 \\ kC-1 & \ldots & kC-1 & 0 & 2kC \end{array}\right) \text{ and } \boldsymbol{v}'_0 = (1, 0, \ldots, 0)^T.$$

Then, with $w = w_1 \ldots w_n$, we can follow that

$$B_w = B_{w_n} \ldots B_{w_2} B_{w_1} = \frac{1}{2(kC)^n} \left(\begin{array}{ccc|cc} & & & 0 & 0 \\ & 2A_w & & \vdots & \vdots \\ & & & 0 & 0 \\ \hline (kC)^n-1 & \ldots & (kC)^n-1 & 2(kC)^n & 0 \\ (kC)^n-1 & \ldots & (kC)^n-1 & 0 & 2(kC)^n \end{array}\right),$$

which gives the final values of the states:

$$\boldsymbol{v}'_f = B_w \boldsymbol{v}'_0 = \frac{1}{(kC)^n} \begin{pmatrix} \vdots \\ \boldsymbol{v}_f \\ \vdots \\ \frac{(kC)^n-1}{2} \\ \frac{(kC)^n-1}{2} \end{pmatrix}.$$

Since $|(\boldsymbol{v}_f)_i| \leq k^{n-1}C^n$, it is clear that $|(\boldsymbol{v}'_f)_i| \leq [-1,1]$: the values of the states are bounded. Now, one has

$$f_{\mathcal{B}} = \frac{|PA_w\boldsymbol{v}_0| + \frac{(kC)^n-1}{2}}{|A_w\boldsymbol{v}_0| + (kC)^n - 1},$$

and so,

$$\begin{aligned} w \in L \Leftrightarrow f_{\mathcal{A}} > \frac{1}{2} &\Leftrightarrow |PA_w\boldsymbol{v}_0| > \frac{1}{2}|A_w\boldsymbol{v}_0| \\ &\Leftrightarrow |PA_w\boldsymbol{v}_0| + \frac{(kC)^n-1}{2} > \frac{1}{2}\left(|A_w\boldsymbol{v}_0| + (kC)^n - 1\right) \\ &\Leftrightarrow f_{\mathcal{B}} > \frac{1}{2}. \end{aligned}$$

□

5 The first languages shown to be not in BAfL

This part is dedicated to prove that some languages are not recognizable by some affine automaton. This is an adaptation of the proof of Turakainen [13] for non-stochastic languages. All the difficulty of exhibiting a non-affine language relies in the fact that a large majority of non-stochasticity proof are based on the linearity of the automaton, which is not the case in the affine case. This proof however, is more based on some "regularity" induced by the matrix-based operations, and number theoric properties of languages like *Prime*. Hence it was possible to adapt it for the affine case, where the only non-linear operation is the final projection.

Let $L \subseteq a^*$ be a unary language. We call *lower density* of L the limit

$$\underline{dens}(L) = \liminf_{n\to\infty} \frac{\left|\{a^k \in L \mid k \leq n\}\right|}{n+1}.$$

Let $(\boldsymbol{x}_n)$ be a sequence of vectors in $\mathbb{R}^k$ and $I = [a_1, b_1) \times \cdots \times [a_k, b_k)$ be an "interval". We define $C(I, n)$ as $C(I, n) = |\{\boldsymbol{x}_i \mod 1 \in I \mid 1 \leq i \leq n\}|$.

We say that $(\boldsymbol{x}_n)$ *is uniformly distributed mod 1* if and only if for any I of such type,

$$\lim_{n\to\infty} \frac{C(I,n)}{n} = (b_1 - a_1)\dots(b_k - a_k).$$

Proposition 8. *If $L \subseteq a^*$ satisfies the following conditions:*

1. *$\underline{dens}(L) = 0$.*
2. *For all $Q \in \mathbb{N}^*$, there exists $h \in \mathbb{N}$ and an infinite sequence $(n_i) \in \mathbb{N}^{\mathbb{N}}$ such that $a^{h+n_iQ} \subseteq L$ and for any irrational number α, the sequence $((h + n_iQ)\alpha)_{i\in\mathbb{N}}$ is uniformly distributed mod 1.*

Then $L \notin$ BAfL.

Proof. Let's assume for contradiction that $L \in$ BAfL. Then there exists an affine automaton A with s states such that

$$f_A(a^n) = \frac{|PM^n\boldsymbol{v}|}{|M^n\boldsymbol{v}|}$$

and there exists $\varepsilon > 0$ such that
- $\forall w \in L, f_A(w) \geq 1 - \varepsilon$,
- $\forall w \notin L, f_A(w) \leq \varepsilon$.

Note that

$$|M^n\boldsymbol{v}| = \sum_{i=1}^{s} |(M^n\boldsymbol{v})_i| \geq \left|\sum_{i=1}^{s} (M^n\boldsymbol{v})_i\right| = 1 \text{ (triangle inequality).}$$

Hence the denominator of f_A is never 0, and so f_A is continuous.

Using the Jordan decomposition $M = PJP^{-1}$, one has $M^n = PJ^nP^{-1}$. So the coordinates $\boldsymbol{v}_i$ of $M^n\boldsymbol{v}$ have the form

$$\boldsymbol{v}_i = \sum_{k=1}^{s} p_{ik}(n)\lambda_k^n \quad (4)$$

where λ_i are the eigenvalues of $\boldsymbol{M}$ and p_{ik} are polynomials of degree less than the degree of the corresponding eigenvalue. Let $\lambda_i = |\lambda_i| e^{2i\pi\theta_i}$, we assume $|\lambda_1| = \cdots = |\lambda_{s'}| > |\lambda_{s'+1}| \ldots$. Let $\lambda = |\lambda_1|$ be the largest module of all eigenvalues and r be the maximum degree of all polynomials p_{ik}, where $k \leq s'$. Then, one can use (4) to write

$$|M^n \boldsymbol{v}| = \sum_{i\in E} |\boldsymbol{v}_i| = \lambda^n n^r \left(\sum_{i\in E} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right| + g_E(n) \right)$$

where a_{ik} is the coefficient of degree r of p_{ik} (note that one can have $a_{ik} = 0$ for some a, k), and g_E a function such that $\lim_{n\to\infty} g_E(n) = 0$. Similarly,

$$|PM^n \boldsymbol{v}| = \sum_{i\in E_a} |\boldsymbol{v}_i| = \lambda^n n^r \left(\sum_{i\in E_a} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right| + g_{E_a}(n) \right).$$

Now let $F(n) = f(a^n)$. Using the previous equations, one has

$$\begin{aligned} F(n) &= \frac{|PM^n \boldsymbol{v}|}{|M^n \boldsymbol{v}|} \\ &= \frac{\lambda^n n^r \left(\sum_{i\in E_a} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right| + g_{E_a}(n) \right)}{\lambda^n n^r \left(\sum_{i\in E} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right| + g_E(n) \right)} \\ &= \frac{\sum_{i\in E_a} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right| + g_{E_a}(n)}{\sum_{i\in E} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right| + g_E(n)}. \end{aligned}$$

We define

$$G(n) = \frac{\sum_{i\in E_a} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right|}{\sum_{i\in E} \left| \sum_{k=1}^{s'} a_{ik} e^{2i\pi n\theta_k} \right|}.$$

As $\lim_{n\to\infty} g_{E_a}(n) = 0$ and $\lim_{n\to\infty} g_E(n) = 0$, one has $G(n) \sim F(n)$, and so,

$$\lim_{n\to\infty} |F(n) - G(n)| = 0. \tag{5}$$

We define $A = \{k \mid 1 \leq k \leq s', \theta_k \notin \mathbb{Q}\}$ the indices of the "first" eigenvalue angles that are not rational. Let Q, h and the sequence (n_i) be as in the statement. Using the periodic behaviour induced by rational angle of eigenvalues, and by taking a subsequence of the initial one, one can also assume that (n_i) is such that

$$G(h + n_i Q) = \frac{\sum_{i\in E_a} \left| \sum_{k\in A} a_{ik} e^{2i\pi(h+n_iQ)\theta_k} + c \right|}{\sum_{i\in E} \left| \sum_{k\in A} a_{ik} e^{2i\pi(h+n_iQ)\theta_k} + d \right|}$$

with c, d some constants.

By assumption, for all $k \in A$, the sequence $((h + n_i Q)\theta_k)_i$ is uniformly distributed modulo 1. The consequence is that the values $e^{2i\pi(h+n_iQ)\theta_k}$ are dense in the unit circle. If for some n, $G(h + nQ) < \frac{1}{2}$, there exists $\varepsilon > 0$ such that

$G(h+nQ) \leq \frac{1}{2} - \varepsilon$. Then, thanks to the density argument, there are arbitrarily large values of i for which $G(h+n_iQ) \leq \frac{1}{2} - \frac{\varepsilon}{2}$. Since for i sufficiently large, $|F(h+n_iQ) - G(h+n_iQ)| \leq \frac{\varepsilon}{2}$ (using (5)), one has $F(h+n_iQ) \leq \frac{1}{2}$, and so $a^{h+n_iQ} \notin L$, contradicting condition 2 of the statement.

Therefore, $G(h+nQ) \geq \frac{1}{2}$ for large enough n. Because G is not identically equal to $\frac{1}{2}$ (if it is the case, F would be as close to $\frac{1}{2}$ as one wants, which is impossible since $L \in \mathsf{BAfL}$), again using density, there must be some $\varepsilon > 0$ and k_0 such that $G(h+k_0Q) \geq \frac{1}{2} + \epsilon$.

First if $A = \emptyset$, it means that all the angles of the eigenvalues $\theta_1, \ldots, \theta_{s'}$ are rational. We can then write them as $\theta_k = \frac{l_k}{m_k}$. Then $G(n)$ takes a finite number of values, and these values only depend on $(n \mod m_1), \ldots, (n \mod m_{s'})$. Let's call $k_1 = h + k_0Q$ the number where G is larger than $\frac{1}{2}$: $G(n_1) > \frac{1}{2}$. G has the same value for all $n \in Z = \{k_1 + km_1 \ldots m_{s'} | k \in \mathbb{N}\}$ (because for n in this set, the values of all $(n \mod m_1), \ldots, (n \mod m_{s'})$ are the same). Then, thanks to (5), one has, for $n \in Z$ sufficiently large, $F(n) > \frac{1}{2}$, so $\{a^n \mid n \in Z, n \geq n_1\} \subseteq L$. And because $|\{a^n \mid n \in Z, n \geq n_1\}| \sim \frac{n}{m_1 \ldots m_{s'}}$, one has $\underline{dens}(L) > 0$, which contradicts condition 1 of the statement.

Next, if $A \neq \emptyset$. Let

$$R((x_k)_{k \in A}) = \frac{\sum_{i \in E_a} \left| \sum_{k \in A} a_{ik} x_k + c \right|}{\sum_{i \in E} \left| \sum_{k \in A} a_{ik} x_k + d \right|}.$$

Note that $G(h+n_iQ) = R((e^{2i\pi(h+n_iQ)\theta_k})_{k \in A})$. Then, because the sequences $((h+n_iQ)\theta_k)_i$ are uniformly distributed modulo 1, it follows that any value obtained by the function $R((e^{2i\pi y_k})_{k \in A})$ can be approximated by some $G(h+n_iQ)$ with arbitrary precision. The function R is continuous, therefore there exists an interval $I = (x_1, y_1, \ldots) = ((x_k, y_k))_{k \in A}$ on which $R((x_k)) > \frac{1}{2} + \frac{\varepsilon}{2}$. So, if n_i is large enough and satisfies

$$((h+n_iQ)\theta_1 \mod 1, \ldots) = ((h+n_iQ)\theta_k \mod 1)_{k \in A} \in I,$$

then $G(h+n_iQ) > \frac{1}{2} + \frac{\varepsilon}{2}$, which implies $F(h+n_iQ) > \frac{1}{2}$ and hence $a^{h+n_iQ} \in L$. Now we just have to prove that the sequence $(h+n_iQ)$ is "dense enough" to have $\underline{dens}(L) > 0$, contradicting again condition 1.
Then, because of uniform distribution imposed by condition 2, one has

$$d = \lim_{i \to \infty} \frac{C(I, h+n_iQ)}{h+n_iQ} = \prod_{k \in A} (y_k - x_k)$$

And so for i large enough, $\frac{C(I,h+n_iQ)}{h+n_iQ} \geq \frac{d}{2}$, with $a^{h+n_iQ} \in L$, implying $\underline{dens}(L) > 0$. We have proved that L cannot be affine. □

Turakainen [13] proved that $Prime = \{a^p \mid p \text{ is prime}\}$ and $Poly(p) = \{a^{p(n)} \mid n \in \mathbb{N}, p(n) \geq 0\}$ satisfy the two conditions of Proposition 8 and they are not in BAfL, where p is any polynomial ($deg > 2$) with non-negative coefficients.

Corollary 2 *$Prime \notin \mathsf{BAfL}$ and $Poly(p) \notin \mathsf{BAfL}$.*

6 Conclusion

In this paper we demonstrated that even if they are strictly more powerful, bounded-error languages of affine automata share stability properties with regular languages (which are bounded-error languages of stochastic automata).

We also showed that the computational power of affine automata does not come alone from the unboundedness state vector set: he general model of unbounded state vector set can always be simulated with a bounded state vector set. Hence it appears obvious that the nonlinear nature of the final value incorporates some computational power, at least in the case of unbounded-error computation.

References

1. Andris Ambainis and John Watrous. Two-way finite automata with quantum and classical states. *Theoretical Computer Science*, 287(1):299–311, sep 2002.
2. A. Ambainis, M. Beaudry, M. Golovkins, A. Ķikusts, M. Mercer, and D. Thérien. Algebraic results on quantum automata. Theory of Computing Systems, 39(1):165—188, 2006.
3. Andris Ambainis and Abuzer Yakaryilmaz. Automata and Quantum Computing. *CoRR*, abs/1507.0:1–32, 2015.
4. Aleksandrs Belovs, Juan Andrés Montoya, and Abuzer Yakaryilmaz. Can one quantum bit separate any pair of words with zero-error? *Tech. Rep.*, 1602.07967, arXiv, 2016.
5. Alejandro Díaz-Caro and Abuzer Yakaryilmaz. Affine computation and affine automaton. In *Computer Science - Theory and Applications - 11th International Computer Science Symposium in Russia, CSR 2016, St. Petersburg, Russia, June 9-13, 2016, Proceedings*, pages 146–160, 2016.
6. Emmanuel Jeandel. Topological automata. *Theory of Computing Systems*, 40(4):397–407, 2007.
7. Attila Kondacs and John Watrous. On the power of quantum finite state automata In FOCS, pages 66-–75. IEEE, 1997
8. Ioan I Macarie. Space-Efficient Deterministic Simulation of Probabilistic Automata. *SIAM Journal on Computing*, 27(2):448–465, 1998.
9. Azaria Paz. *Introduction to Probabilistic Automata (Computer Science and Applied Mathematics)*. Academic Press, Inc., Orlando, FL, USA, 1971.
10. M. O. Rabin. Probabilistic automata. Information and Control, 6:230-–243, 1963.
11. Michael Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1st edition, 1996.
12. Paavo Turakainen. Generalized Automata and Stochastic Languages. *Proceedings of the American Mathematical Society*, 21(2):303–309, 1969.
13. Paavo Turakainen. On nonstochastic languages and homomorphic images of stochastic languages. *Information Sciences*, 24(3):229–253, aug 1981.
14. Marcos Villagra and Abuzer Yakaryilmaz. *Language Recognition Power and Succinctness of Affine Automata*, pages 116–129. Springer International Publishing, Cham, 2016.