Computational Limitations of Affine Automata

Mika Hirvensalo, Etienne Moutot, Abuzer Yakaryilmaz

Computational Limitations of Affine Automata

Mika Hirvensalo[1], Etienne Moutot[1,2][0000−0003−2073−4709], and Abuzer Yakaryılmaz[3][0000−0002−2372−252X]

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

[2] LIP, ENS de Lyon CNRS UCBL Université de Lyon , École Normale Supérieure de Lyon, Lyon, France
etienne.moutot@ens-lyon.org

[3] Center for Quantum Computer Science, Faculty of Computing University of Latvia, Rīga, Latvia
abuzer@lu.lv

Abstract. We present two new results on the computational limitations of affine automata. First, we show that the computation of bounded-error rational-values affine automata is simulated in logarithmic space. Second, we give an impossibility result for algebraic-valued affine automata. As a result, we identify some unary languages (in logarithmic space) that are not recognized by algebraic-valued affine automata with cutpoints.

1 Introduction

Finite automata are an interesting model to study since they express the very natural limitation of finite memory. They are also good computational models, since they are simpler than many others machines like pushdown automata or Turing machines. Due to this simplicity, there exists many different models of finite automata, all trying to express different computational settings. Deterministic [16], probabilistic [14] and quantum [3] finite automata (DFAs, PFAs, and QFAs, respectively) have been studied to try to understand better the computational limitations inherent to all these cases.

Recently, Díaz-Caro and Yakaryılmaz introduced a new model, called *affine computation* [5]. As a non-physical model, the goal of affine computation is to investigate the power of interference caused by negative amplitudes in the computation, like in the quantum case. But unlike QFAs, affine finite automata (AfAs) have unbounded state set and the final operation corresponding to quantum measurement cannot be interpreted as linear. The final operation in AfAs is analogous to renormalization in Kondacs-Watrous [11] or Latvian [2] quantum automata models.

AfAs and their certain generalizations have been investigated in a series of works [5, 8, 9, 21]. In most of the cases, affine models (e.g., bounded-error and unbouded-error AfAs, zero-error affine OBDDs, zero-error affine counter automata, etc.) have been shown more powerful than their classical or quantum

counterparts. On the other hand, we still do not know too much regarding the computational limitations of AfAs. Towards this direction, we present two new results. First, we show that the computation of bounded-error rational-values affine automata is simulated in logarithmic space, and so we answer positively one of the open problems in [5]. Second, we give an impossibility result for algebraic-valued AfAs, and, as a result, we identify some unary languages (in logarithmic space) that are not recognized by algebraic-valued AfAs with cut-points.

2 Preliminaries

For a given word w, w_i represents its i-th letter. For any given class C, $\mathsf{C}_{\mathbb{Q}}$ and $\mathsf{C}_{\mathbb{A}}$ denotes the classes defined by the machines restricted to have rational-valued and algebraic-valued components, respectively. The logarithmic and polynomial space classes are denoted as L and PSPACE, respectively. We assume that the reader is familiar with the basics of automata theory.

2.1 Models

As a *probability distribution* (also known as a *stochastic vector*) we understand a (column) vector with nonnegative entries summing up to one, and a *stochastic matrix* (also known as a *Markov matrix*) here stands for a square matrix whose all columns are probability distributions.

Definition 1 (PFA). *A k-state probabilistic finite automaton (PFA) P over alphabet Σ is a triplet*

$$P = (\boldsymbol{x}, \{M_i \mid i \in \Sigma\}, \boldsymbol{y})$$

where $\boldsymbol{x} \in \mathbb{R}^k$ is a stochastic vector called initial distribution*, each $M_i \in \mathbb{R}^{k\times k}$ is a stochastic matrix, and $\boldsymbol{y} \in \{0,1\}^k$ is the final vector (each 1 in $\boldsymbol{y}$ represents an accepting state).*

For any input word $w \in \Sigma^*$ with length n, P has a probability distribution of states as follows: $M_w\boldsymbol{x} = M_{w_n} \cdots M_{w_1}\boldsymbol{x}$. The *accepting probability* corresponds to the probability of P being in an accepting state after reading w, which is given by

$$f_P(w) = \boldsymbol{y}^T M_{w^R}\boldsymbol{x}. \tag{1}$$

Affine finite automaton (AfA) is a generalization of PFA allowing negative transition values. Only allowing negative values in the transition matrices does not add any power (generalized PFAs are equivalent to PFAs, see [19]), but affine automata introduce also a non-linear behaviour. The automaton acts like a generalized probabilistic automaton until the last operation, which is a non-linear operation called a *weighting operation*.

Definition 2. *A vector $\boldsymbol{v} \in \mathbb{R}^k$ 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.*

The following property is straightforward to verify, and it will ensure that affine automata are well defined.

Property 1 *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.*

Definition 3 (AfA). *A k-state* AfA *A over alphabet Σ is a triplet*

$$A = (\boldsymbol{x}, \{M_i \mid i \in \Sigma\}, F)$$

where $\boldsymbol{x}$ is an initial affine vector, each M_i is an affine transition matrix, and $F = \mathrm{diag}(\delta_1, \ldots, \delta_n)$ is the final projection matrix, where each $\delta_i \in \{0, 1\}$ for $1 \leq i \leq n$.

The value computed by an affine automaton can be most conveniently be defined via the following notion:

Definition 4. *Notation $|\boldsymbol{v}| = \sum_i |v_i|$ stands for the usual L^1 norm.*

Now, the final value of the affine automaton A of Definition 3 is

$$f_A(w) = \frac{|FM_w\boldsymbol{v}_0|}{|M_w\boldsymbol{v}_0|}. \tag{2}$$

Clearly $f_A(w) \in [0, 1]$ for any input word $w \in \Sigma^*$.

Remark 1. Notice that the final value for PFAs (1) is defined as matrix product $\boldsymbol{v}_f \mapsto \boldsymbol{y}^T\boldsymbol{v}_f$, which is a linear operation on $\boldsymbol{v}_f$. On the other hand, computing final value from $\boldsymbol{v}_f$ as in (2) involves nonlinear operations $\boldsymbol{v}_f \mapsto \dfrac{|F\boldsymbol{v}_f|}{|\boldsymbol{v}_f|}$ such as L^1-norm and normalization (division).

2.2 Cutpoint languages

Given a function $f : \Sigma^* \to [0, 1]$ computed by an automaton (stochastic or affine), there are different ways of defining the language of recognized by this automaton.

Definition 5 (Cutpoint languages). *A language $L \subseteq \Sigma^*$ is recognized by an automaton A with cutpoint $\lambda \in [0, 1)$ 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*).*

We remark that fixing the cutpoint in the interval $(0,1)$ does not change the classes SL and AfL [5,14].

Definition 6 (Exclusive cutpoint languages). *A language $L \subseteq \Sigma^*$ is recognized by an automaton A with exclusive cutpoint $\lambda \in [0,1]$ if and only if*

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

These languages are called exclusive cutpoint languages. In the case of probabilistic (resp., affine automata), the set of exclusive cut-point languages are called exclusive stochastic languages *(resp.,* exclusive affine languages*) and denoted by $\mathsf{SL}^{\neq}$ (resp., $\mathsf{AfL}^{\neq}$). The complement of $\mathsf{SL}^{\neq}$ (resp., $\mathsf{AfL}^{\neq}$) is $\mathsf{SL}^{=}$ (resp., $\mathsf{AfL}^{=}$).*

Again, we remark that fixing the cutpoint in the interval $(0,1)$ does not change the classes $\mathsf{SL}^{\neq}$, $\mathsf{SL}^{=}$, $\mathsf{AfL}^{\neq}$, and $\mathsf{AfL}^{=}$ [5,13,14].

A stronger condition is to impose that accepted and rejected words are separated by a gap: the cutpoint is said to be isolated.

Definition 7 (Isolated cutpoint or bounded error). *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 $\forall w \notin L, f_A(w) \leq \lambda - \delta$. The set of languages recognized with* bounded error *(or isolated cutpoint) affine automata is denoted by BAfL.*

A classical result by Rabin [15] shows that isolated cutpoint stochastic languages are regular. 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 [3]: 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 [10].

3 Logarithmic simulation

Macarie [12] proved that $\mathsf{SL}^{=}_{\mathbb{Q}} \subseteq \mathsf{L}$ and $\mathsf{SL}_{\mathbb{Q}} \subseteq \mathsf{L}$. That is, the computation of any rational-valued probabilistic automaton can be simulated by an algorithm using only logarithmic space. However, this logarithmic simulation cannot be directly generalized for rational-valued affine automata due to the non-linearity of their last operation. In order to understand why, we will first reproduce the proof.

Before that, let us introduce the most important space-saving technique:

Definition 8. *Notation $(b \mod c)$ stands for the least nonnegative integer a satisfying $a \equiv b \pmod{c}$. If $\boldsymbol{x} = (x_1, \ldots, x_r)$ and $\boldsymbol{n} = (n_1, \ldots, n_r) \in \mathbb{Z}^r$, we define $\boldsymbol{x} \pmod{\boldsymbol{n}} = ((x_1 \bmod n_1), \ldots, (x_r \bmod n_r))$. Analogously, for any matrix $A \in \mathbb{Z}^{k \times k}$, we define $(A \pmod n)_{ij} = (A_{ij} \mod n)$.*

The problem of recovering x from the residue representation $((x \bmod n_1), \ldots, (x \bmod n_r))$ is practically resolved by the following well-known theorem.

Theorem 2 (The Chinese Remainder Theorem). *Let $n_1, \ldots, n_r$ be pairwise coprime integers, $a_1, \ldots, a_r$ be arbitrary integers, and $N = n_1 \cdots n_r$. Then there exists an integer x such that*

$$x \equiv a_1 \pmod{n_1}, \ldots, x \equiv a_r \pmod{n_r}, \tag{3}$$

and any two integers x_1 and x_2 satisfying (3) satisfy also $x_1 \equiv x_2 \pmod{N}$.

Remark 2. The above remarks and the Chinese Remainder Theorem imply that the integer ring operations $(+, \cdot)$ can be implemented using the residue representation, and that the integers can be uncovered from the residue representations provided that 1) $\boldsymbol{n} = (n_1, \ldots, n_r)$ consists of pairwise coprime integers and 2) the integers stay in interval of length $N - 1$, where $N = n_1 \cdots n_r$.

Remark 3. In order to ensure that $\boldsymbol{n} = (n_1, \ldots, n_r)$ consists of pairwise coprime integers, we select numbers n_i from the set of prime numbers. For the reasons that will become obvious later, we will however omit the first prime 2.

Definition 9. *$\boldsymbol{p}_r$ is an r-tuple $\boldsymbol{p}_r = (3, 5, 7, \ldots, p_r)$ consisting of r first primes by excluding 2. For this selection, a consequence of the prime number theorem is that, asymptotically, $P_r = 3 \cdot 5 \cdot 7 \cdot \cdots \cdot p_r = \frac{1}{2} e^{(1+o(1)) r \ln r}$.*

Theorem 3 (Macarie [12]). $\mathsf{SL}_{\mathbb{Q}}^{=} \subseteq \mathsf{L}$

Proof. For a given alphabet Σ, let $L \in \Sigma^*$ be a language in $\mathsf{SL}_{\mathbb{Q}}^{=}$ and $P = (\boldsymbol{x}, \{M_i \mid i \in \Sigma\}, \boldsymbol{y})$ be a k-state rational-valued PFA over Σ such that

$$L = \left\{ w \in \Sigma^* \mid f_P(w) = \frac{1}{2} \right\}.$$

We remind that, for any input word $w = w_1 \cdots w_n \in \Sigma^*$, we have

$$f_P(w) = \boldsymbol{y}^T M_{w_n} \cdots M_{w_1} \boldsymbol{x}. \tag{4}$$

Since each $M_i \in \mathbb{Q}^{k \times k}$, there exists a number $D \in \mathbb{N}$ providing that each matrix $M_i' = DM_i \in \mathbb{Z}^{k \times k}$, and (4) can be rewritten as

$$f_P(w) = \frac{1}{D^n} \underbrace{\boldsymbol{y}^T M_{w_n}' \ldots M_{w_1}' \boldsymbol{x}}_{f_{P'}(w)},$$

and the language L can be characterized as

$$L = \{ w \in \Sigma^* \mid 2 f_{P'}(w) = D^n \}. \tag{5}$$

Since the original matrices M_i are stochastic, meaning that their entries are in $[0, 1]$, it follows that each matrix $M_i' = DM_i$ has integer entries in $[0, D]$. Moreover, $f_P(w) \in [0, 1]$ implies that $f_{P'}(w) \in [0, D^n]$ for every input word $w \in \Sigma^n$. As now $f_{P'}(w)$ can be computed by multiplying $k \times k$ integer matrices, the residue representation will serve as a space-saving technique.

We will fix r later, but the description of the algorithm is as follows: For each entry p of $\boldsymbol{p}_r = (3, 5, 7, \ldots, p_r)$, we let $M_i^{(p)} = M_i' \mod p$, and compute

$$(2f_{P'}(w) \mod p) = \boldsymbol{y}^T M_{w_n}^{(p)} \cdots M_{w_1}^{(p)} \boldsymbol{x} \tag{6}$$

as all the products are computed modulo p, $k^2 \log p$ bits are needed to compute (6). Likewise, $(D^n \bmod p)$ can be computed in space $O(\log p)$ for each coordinate p of $\boldsymbol{p}_r$. The comparison $2f_{P'}(w) \equiv D^n \pmod p$ can hence done in $O(\log p)$ space.

Reusing the space, the comparison can be made sequentially for each coordinate of $\boldsymbol{p}_r$, and if any comparison gives a negative outcome, we can conclude that $2P'(w) \neq D^n$.

To conclude the proof, it remains to fix r so that both $2f_{P'}(w)$ and D^n are smaller than $P_r = 3 \cdot 5 \cdot 7 \cdot \cdots \cdot p_r$. If no congruence test is negative, then the Chinese Remainder Theorem ensures that $2f_{P'}(w) = D^n$. Since $2f_{P'}(w) \leq D^n$, we need to select r so that $\frac{1}{2}e^{(1+o(1))r \ln r} > 2D^n$, which is equivalent to $\log \frac{1}{2} + (1 + o(1))r \ln r > \log 2 + n \log D$. This inequality is clearly satisfied with $r = n$ for large enough n, and for each $n \geq 1$ by choosing $r = c \cdot n$, where c is a positive constant (depending on D).

As a final remark let us note that $p_{\lfloor cn \rfloor}$, the $\lfloor cn \rfloor$-th prime, can be generated in logarithmic space and the prime number theorem implies that $O(\log n)$ bits are enough to present $p_{\lfloor cn \rfloor}$, since c is a constant. □

To extend the above theorem to cover $\mathsf{SL}_{\mathbb{Q}}$ as well, auxiliary results are used.

Lemma 1 (Macarie [12]). *If N is an odd integer and x, $y \in [0, N-1]$ are also integers, then $x \geq y$ iff $x - y$ has the same parity as $((x - y) \bmod N)$.*

Proof. As x, $y \in [0, N-1]$, it follows that

$$(x - y \mod N) = \begin{cases} x - y \text{ if } x \geq y \\ N + x - y \text{ if } x < y, \end{cases}$$

which shows that the parity changes in the latter case since N is odd. □

The problem of using the above lemma is that, in modular computing, numbers x and y are usually known only by their residue representations $\mathrm{Res}_{\boldsymbol{p}_r}(x)$ and $\mathrm{Res}_{\boldsymbol{p}_r}(y)$, and it is not straightforward to compute the parity from the modular representation in logarithmic space. Macarie solved this problem not only for parity but also for a more general modulus (not necessarily equal to 2).

Lemma 2 (Claim modified from [12]). *For any integer x and modulus $\boldsymbol{p}_r = (3, 5, 7, \ldots, p_r)$, there is a deterministic algorithm that given $\mathrm{Res}_{\boldsymbol{p}_r}(x)$ and $M \in \mathbb{Z}$ as input, produces the output $x \pmod M$ in space $O(\log p_r + \log M)$*

As a corollary of the previous lemma, Macarie presented a conclusion which implies the logarithmic space simulation of rational stochastic automata.

Lemma 3 (Claim modified from [12]). *Let $\boldsymbol{p}_r = (3, 5, 7, \ldots, p_r)$ and $P_r = 3 \cdot 5 \cdot 7 \cdot \cdots \cdot p_r$. Given the residue representations of integers x, $y \in [0, P_r - 1]$, the decisions $x > y$, $x = y$ or $x < y$ can be made in $O(\log p_r)$ space.*

Proof. The equality test can be done as in the proof Theorem 3, testing the congruence sequentially for each prime. Testing $x \geq y$ is possible by lemmata 1 and 2: First compute $\mathrm{Res}_{\boldsymbol{p}_r}(z) = \mathrm{Res}_{\boldsymbol{p}_r}(x) - \mathrm{Res}_{\boldsymbol{p}_r}(y) \pmod{\boldsymbol{p}_r}$, then compute the parities of x, y, z using Lemma 2 with $M = 2$. □

The following theorem is a straightforward corollary from the above:

Theorem 4. $\mathsf{SL}_{\mathbb{Q}} \subseteq \mathsf{L}$.

When attempting to prove an analogous result to affine automata, there is at least one obstacle: computing the final value includes the absolute values, but the absolute value is not even a well-defined operation in the modular arithmetic. For example, $2 \equiv -3 \pmod 5$, but $|2| \not\equiv |-3| \pmod 5$. This is actually another way to point out that, in the finite fields, there is no order relation compatible with the algebraic structure.

Hence for affine automata with matrix entries of both signs, another approach must be adopted. One obvious approach is to present an integer n as a pair $(|n|, \mathrm{sgn}(n))$, and apply modular arithmetic to $|n|$. The signum function and the absolute value indeed behave smoothly with respect to the product, but not with the sum, which is a major problem with this approach, since to decide the sign of the sum requires a comparison of the absolute values, which seems impossible without having the whole residue representation. The latter, in its turn seems to cost too much space resources to fit the simulation in logarithmic space.

Hence the logspace simulation for automata with matrices having both positive and negative entries seems to need another approach. It turns out that we can use the procedure introduced by Turakainen already in 1969 [17, 19].

Theorem 5. $\mathsf{AfL}_{\mathbb{Q}} \subseteq \mathsf{L}$.

Proof. For a given alphabet Σ, let $L \in \Sigma^*$ be a language in $\mathsf{AfL}_{\mathbb{Q}}$ and $A = (\boldsymbol{x}, \{M_i \mid i \in \Sigma\}, F)$ be a k-state rational-valued AfA over Σ such that

$$L = \left\{ w \in \Sigma^* \mid f_A(w) > \frac{1}{2} \right\}.$$

For each $M_i \in \mathbb{Q}^{k \times k}$, we define a new matrix as $B_i = \begin{pmatrix} 0 & \mathbf{0}^T & 0 \\ \boldsymbol{c}_i & M_i & \mathbf{0} \\ e_i & \boldsymbol{d}_i^T & 0 \end{pmatrix}$, where $\boldsymbol{c}_i$, $\boldsymbol{d}_i$, and e_i are chosen so that the column and row sums of B_i are zero. We define $\boldsymbol{x}' = \begin{pmatrix} 0 \\ \boldsymbol{x} \\ 0 \end{pmatrix}$ as the new initial state. For the projection matrix F, we define an extension $F' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & 0 \end{pmatrix}$. It is straightforward to see that $|B_w \boldsymbol{v}_0'| = |M_w v_0|$ as well as $|F' B_w \boldsymbol{v}_0'| = |F M_w v_0|$.

For the next step, we introduce a $(k+2) \times (k+2)$ matrix $\mathbb{E}$, whose each element is 1. It is then clear that $\mathbb{E}^n = (k+2)^{n-1}\mathbb{E}$ and $B_i \mathbb{E} = \mathbb{E} B_i = \mathbf{0}$. Now

we define

$$C_i = B_i + m\mathbb{E},$$

where $m \in \mathbb{Z}$ is selected large enough to ensure the nonnegativity of the matrix entries of each C_i. It follows that

$$C_w = B_w + m^{|w|}(k+2)^{|w|-1}\mathbb{E},$$

and

$$C_w\boldsymbol{x}' = B_w\boldsymbol{x}' + m^{|w|}(k+2)^{|w|-1}\mathbb{E}\boldsymbol{x}'.$$

Similarly,

$$F'C_w\boldsymbol{x}' = F'B_w\boldsymbol{x}' + m^{|w|}(k+2)^{|w|-1}F'\mathbb{E}\boldsymbol{x}'.$$

Now

$$\frac{|FM_w\boldsymbol{v}_0|}{|M_w\boldsymbol{v}_0|} = \frac{|F'B_w\boldsymbol{v}_0|}{|B_w\boldsymbol{v}_0|} = \frac{\left|F'C_w\boldsymbol{v}_0' - m^{|w|}(k+2)^{|w|-1}F'\mathbb{E}\boldsymbol{x}'\right|}{\left|C_w\boldsymbol{x}' - m^{|w|}(k+2)^{|w|-1}\mathbb{E}\boldsymbol{x}'\right|}$$

which can further be modified by expanding the denominators away: For an integer g large enough all matrices $D_i = gC_i$ will be integer matrices and the former equation becomes

$$\frac{|FM_w\boldsymbol{x}|}{|M_w\boldsymbol{x}|} = \frac{|F'B_w\boldsymbol{x}|}{|B_w\boldsymbol{x}|} = \frac{\left|F'D_w\boldsymbol{x}' - m^{|w|}(k+2)^{|w|-1}g^{|w|+1}F'\mathbb{E}\boldsymbol{x}'\right|}{\left|D_w\boldsymbol{x}' - m^{|w|}(k+2)^{|w|-1}g^{|w|+1}\mathbb{E}\boldsymbol{x}'\right|}. \tag{7}$$

Hence the inequality

$$\frac{|FM_w\boldsymbol{x}|}{|M_w\boldsymbol{x}|} \geq \frac{1}{2}$$

is equivalent to

$$\begin{aligned} &2\left|F'D_w\boldsymbol{x}' - m^{|w|}(k+2)^{|w|-1}g^{|w|+1}F'\mathbb{E}\boldsymbol{x}'\right| \\ &\geq \left|D_w\boldsymbol{x}' - m^{|w|}(k+2)^{|w|-1}g^{|w|+1}\mathbb{E}\boldsymbol{x}'\right|. \end{aligned} \tag{8}$$

In order to verify inequality (8) in logarithmic space, it sufficient to demonstrate that the residue representations of both sides can be obtained in logarithmic space.

For that end, the residue representation of vector $\boldsymbol{a} = F'D_w\boldsymbol{x}' \in \mathbb{R}^{k+2}$ can be obtained in logarithmic space as in the proof of Theorem 3.

Trivially, the residue representation of $\boldsymbol{b} = m^{|w|}(k+2)^{|w|-1}g^{|w|+1}F'\mathbb{E}\boldsymbol{x}' \in \mathbb{R}^{k+2}$ can be found in logarithmic space, as well. In order to compute the residue representation of

$$|\boldsymbol{a} - \boldsymbol{b}| = |\boldsymbol{a}_1 - \boldsymbol{b}_1| + \cdots + |\boldsymbol{a_k} - \boldsymbol{b}_k|$$

it is sufficient to decide whether $\boldsymbol{a}_i \geq \boldsymbol{b}_i$ holds. As the residue representations for each $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ is known, all the decisions can be made in logspace, according to Lemma 3. The same conclusion can be made for the right hand side of (8). □

4 A Non-affine Language

As we saw in the previous section, $\mathsf{AfL}_{\mathbb{Q}} \subseteq \mathsf{L}$, and hence languages beyond L, are good candidates for non-affine languages.[4] In this section, we will however demonstrate that the border of non-affinity may lie considerably lower: There are languages in L which are not affine.

In an earlier work [8], we applied the method of Turakainen [20] to show that there are languages in L which however are not contained in BAfL. Here we will extend the previous result to show that those languages are not contained even in $\mathsf{AfL}_{\mathbb{A}}$. (We leave open whether a similar technique can be applied for AfL.)

Definition 10 (Lower density).
*Let $L \subseteq a^*$ be a unary language. We call **lower density** of L the limit*

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

Definition 11 (Uniformly distributed sequence). *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 in $\mathbb{R}^k$. We define $C(I, n)$ as $C(I, n) = |\{\boldsymbol{x}_i \mod 1 \in I \mid 1 \le i \le n\}|$.*

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

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

Theorem 6. *If $L \subseteq a^*$ satisfies the following conditions:*

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

Then L is not in $\mathsf{AfL}_{\mathbb{A}}$.

Proof. Let's assume for contradiction that $L \in \mathsf{AfL}_{\mathbb{A}}$. Then there exists an AfA A with s states, matrix M and initial vector $\boldsymbol{v}$ such that the acceptance value of A is

$$f_A(a^n) = \frac{|PM^n\boldsymbol{v}|}{|M^n\boldsymbol{v}|}. \tag{9}$$

Without loss of generality, we can assume that the cutpoint equals to $\frac{1}{2}$, and hence $w \in L \Leftrightarrow f_A(w) > \frac{1}{2}$.

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

$$(M^n\boldsymbol{v})_j = \sum_{k=1}^{s} p_{jk}(n)\lambda_k^n, \tag{10}$$

[4] It is known that $\mathsf{L} \subsetneq \mathsf{PSPACE}$, so it is plausible that PSPACE-complete languages are not in $\mathsf{AfL}_{\mathbb{Q}}$.

where λ_k are the eigenvalues of M and p_{jk} are polynomials of degree less than the degree of the corresponding eigenvalue. For short, we denote $F(n) = f_A(a^n)$, and let $\lambda_k = |\lambda_k| e^{2i\pi\theta_k}$.

When studying expression (9), we can assume without loss of generality, that all numbers θ_k are irrational. In fact, replacing matrix M with αM, where $\alpha \neq 0$ does not change (9), since

$$\frac{|P(\alpha M)^n \boldsymbol{v}|}{|(\alpha M)^n \boldsymbol{v}|} = \frac{|\alpha^n P M^n \boldsymbol{v}|}{|\alpha^n M^n \boldsymbol{v}|} = \frac{|PM^n \boldsymbol{v}|}{|M^n \boldsymbol{v}|}.$$

Selecting now $\alpha = e^{2\pi i\theta}$ (where $\theta \in \mathbb{R}$) implies that the eigenvalues of M are $\lambda_k e^{2i\pi(\theta_k+\theta)}$. The field extension $\mathbb{Q}(\theta_1, \ldots, \theta_s)$ is finite, and hence there is always an irrational number $\theta \notin \mathbb{Q}(\theta_1, \ldots, \theta_s)$. It follows directly that all numbers $\theta_k + \theta$ are irrational. Hence we can assume that all the numbers θ_k are irrational in the first place.[5]

By restricting to an arithmetic progression $n = r + mN$ ($m \in \mathbb{N}$) we can also assume that no λ_i/λ_j is a root of unity for $i \neq j$. In fact, selecting $N = \mathrm{lcm}\{\mathrm{ord}(\lambda_i/\lambda_j) \mid i \neq j \text{ and } \lambda_i/\lambda_j \text{ is a root of unity}\}$ (10) becomes

$$(M^{r+mN}\boldsymbol{v})_j = \sum_{k=1}^{s} p_{jk}(r+mN)\lambda_k^r(\lambda_k)^{Nm} = \sum_{k=1}^{s'} q_{jk}(m)\mu_k^m, \tag{11}$$

where $\{\mu_1, \ldots, \mu_{s'}\}$ are the distinct elements of set $\{\lambda_1^N, \ldots, \lambda_s^N\}$ Now for $i \neq j$ μ_i/μ_j cannot be a root of unity, since $(\mu_i/\mu_j)^t = 1$ would imply $(\lambda_{i'}/\lambda_{j'})^{Nt} = 1$, which in turn implies $(\lambda_{i'}/\lambda_{j'})^N = 1$ and hence $\mu_i = \lambda_{i'}^N = \lambda_{j'}^N = \mu_j$, which contradicts the assumption $\mu_i \neq \mu_j$.

We can now write the acceptance condition $f_A(a^n) > \frac{1}{2}$ equivalently as

$$\begin{aligned} & f_A(a^n) > \frac{1}{2} \Leftrightarrow 2\,|PM^n\boldsymbol{v}| > |M^n\boldsymbol{v}| \\ \Leftrightarrow\ & 2\sum_{j\in E_a} |(M^n\boldsymbol{v})_j| > \sum_{j\in E} |(M^n\boldsymbol{v})_j| \Leftrightarrow \underbrace{\sum_{j\in E_a} |(M^n\boldsymbol{v})_j| - \sum_{j\in \overline{E_a}} |(M^n\boldsymbol{v})_j|}_{g(n)} > 0, \end{aligned}$$

Where E is the set of states of A, $E_a \subseteq E$ its set of accepting states, and $\overline{E_a}$ the complement of E_a. According to (10), $g(n) := \sum_{j\in E_a} |(M^n\boldsymbol{v})_j| - \sum_{j\in\overline{E_a}} |(M^n\boldsymbol{v})_j|$ consists of combinations of absolute values of linear combination of functions of type $n^d\lambda^n$.

We say that $n^{d_1}\lambda_1^n$ is of *larger order* than $n^{d_2}\lambda_2^n$, if $|\lambda_1| > |\lambda_2|$; and in the case $|\lambda_1| = |\lambda_2|$, if $d_1 > d_2$. If $|\lambda_1| = |\lambda_2|$, we say that $n^d\lambda_1^n$ and $n^d\lambda_2^n$ and of the same order. It is clear that if term $t_1(n)$ is of larger order than $t_2(n)$, then $\lim_{n\to\infty} \frac{t_2(n)}{t_1(n)} = 0$.

[5] Note that the new matrix obtained may not be affine, so it would be wrong to assume that all AfAs have to admit an equivalent one with only irrational eigenvalues. However, this does not affect this proof, since we do not require the new matrix to be affine, we only study the values that the fraction $\frac{|P(\alpha M)^n\boldsymbol{v}|}{|(\alpha M)^n\boldsymbol{v}|} = \frac{|PM^n\boldsymbol{v}|}{|M^n\boldsymbol{v}|}$ take.

We can organize the terms in expression (10) as

$$(M^n \boldsymbol{v})_j = \sum_{k=1}^{s} p_{jk}(n)\lambda_k^n = \Lambda_j^{(N)}(n) + \Lambda_j^{(N-1)}(n) + \cdots + \Lambda_j^{(0)}(n), \tag{12}$$

where each $\Lambda_j^{(m)}(n)$ consists of terms with equal order multiplier:

$$\Lambda_j^{(m)}(n) = \sum_{k=1}^{m_j} c_{mk} n^{d_m} {\lambda_{mk}}^n = n^{d_m} \lambda_m^n \sum_{k=1}^{m_j} c_{mk} e^{2\pi i n \theta_{mk}} \tag{13}$$

(for notational simplicity, we mostly omit the dependency on j in the right hand side of (13)). Here $\lambda_m \in \mathbb{R}_+$ is the common absolute value of all eigenvalues $\lambda_{mk} = \lambda_m e^{2\pi i \theta_{mk}}$, and expression (12) is organized in descending order: $\Lambda_j^{(N)}$ is the sum of terms of the highest order multiplier, $\Lambda_j^{(N-1)}$ contains the terms of the second highest order multiplier, etc. We say that $\Lambda_j^{(k_2)}$ is lower than $\Lambda_j^{(k_1)}$ if $k_2 < k_1$

We will then fix a representation

$$\begin{aligned} g(n) &= \sum_{j \in E_a} \left| \sum_{k=1}^{s} p_{jk}(n)\lambda_k^n \right| - \sum_{j \in \overline{E_a}} \left| \sum_{k=1}^{s} p_{jk}(n)\lambda_k^n \right| \\ &= \sum_{j \in E_a} |A_j(n) + B_j(n) + C_j(n)| - \sum_{j \in \overline{E_a}} |A_j(n) + B_j(n) + C_j(n)|, \end{aligned} \tag{14}$$

where $A_j(n) + B_j(n) + C_j(n)$ is a grouping of all Λ-terms in (12) defined as follows:

1. $A_j(n) = \sum_{k=0}^{m} \Lambda_j^{(N-k)}(n)$, where $m \in [-1, N] \cap \mathbb{Z}$ is chosen as the maximal number so that
$$A = \sum_{j \in E_a} |A_j(n)| - \sum_{j \in \overline{E_a}} |A_j(n)| \tag{15}$$
is a constant function $\mathbb{N} \to \mathbb{R}$. Such an m exists, since for $m = -1$, the sum is regarded empty and $A_j(n) = 0$, but for $m = N$, all Λ-terms are included, and then (15) becomes $f_A(a^n)$, which is not constant (otherwise condition 1 or 2 of the theorem would be false).
2. $B_j(n)$ consists a single Λ-term immediately lower than those in $A_j(n)$4, and
3. $C_j(n)$ contains the rest of the Λ-terms, lower than $B_j(n)$

Lemma 4. *If $A \neq 0$, then $\forall z \in \mathbb{C}, |A + z| = |A| + \operatorname{Re} \frac{|A|}{A} z + O(\frac{z^2}{A})$.*

Proof. Denote $z = x + iy$. Because $|\operatorname{Re} z| \leq |z|$, we have

$$\begin{aligned} |1 + z| = |1 + x + iy| &= \sqrt{(1+x)^2 + y^2} = \sqrt{1 + 2\operatorname{Re} z + |z|^2} \\ &= 1 + \operatorname{Re} z + O(z^2). \end{aligned}$$

Now

$$|A+z| = |A|\left|1+\frac{z}{A}\right| = |A|\left(1+\operatorname{Re}\frac{z}{A}+O((\frac{z}{A})^2)\right) = |A|+\operatorname{Re}\frac{|A|}{A}z+O(\frac{z^2}{A}).$$

□

We choose $\lambda \in \mathbb{R}_+$ and d so that the highest Λ-term in $B(n)$ is of order $n^d\lambda^n$ and define $A'_j(n) = n^{-d}\lambda^{-n}A_j(n)$, $B'_j(n) = n^{-d}\lambda^{-n}B_j(n)$, $g'(n) = g(n)n^{-d}\lambda^{-n}$. Then clearly $g'(n) > 0$ if and only if $g(n) > 0$ and each $B_j(n)$ remains bounded as $n \to \infty$. To simplify the notations, we omit the primes and recycle the notations to have a new version of $g(n)$ of (14) where A_j-terms may tend to infinity but B_j-terms remain bounded.

Recall that we may assume (by restricting to a arithmetic progression) that no λ_i/λ_j is a root of unity. By Skolem-Mahler-Lech theorem [7], this implies that functions A_j can have only a finite number of zeros, and in the continuation we assume that n is chosen so large that no function A_j becomes zero. Furthermore, by the main theorem of [6], then $|A_j(n)| = \Omega(n^d\lambda^{n-\epsilon})$ for each $\epsilon > 0$.[6] As each B_j remains bounded, we find that B_j^2/A_j tend to zero as $n \to \infty$, and hence by Lemma 4, defining

$$\begin{aligned} &g_1(n) = \\ &\sum_{j\in E_a}\Big(|A_j(n)| + \operatorname{Re}(\frac{|A_j(n)|}{A_j(n)}B_j(n))\Big) - \sum_{j\in\overline{E_a}}\Big(|A_j(n)| + \operatorname{Re}(\frac{|A_j(n)|}{A_j(n)}B_j(n))\Big) \\ &= \underbrace{\sum_{j\in E_a}|A_j(n)| - \sum_{j\in\overline{E_a}}|A_j(n)|}_{h(n)} + \sum_{j\in E_a}\operatorname{Re}(\frac{|A_j(n)|}{A_j(n)}B_j(n)) + \sum_{j\in\overline{E_a}}\operatorname{Re}(\frac{|A_j(n)|}{A_j(n)}B_j(n)) \end{aligned}$$

we have a function $g_1(n)$ with the property $g_1(n) - g(n) \to 0$ (C-terms are lower than B-terms, so they can be dropped without violating this property), when $n \to \infty$. Also by the construction it is clear that $h(n) = C \cdot n^d\lambda^n$, where C is a constant, and by the conditions of the theorem, this is possible only if $C = 0$.

Notice tat $g_1(n)$ is not a constant function by construction. Also, each B_j is a linear combination of functions of form $e^{2\pi i\theta_k n}$, each θ_k can be assumed irrational, and $||A_j(n)|\, A_j(n) = 1|$, so we can conclude that $g_1(n)$ is a continuous function formed of terms of form $ce^{i\theta_k n}$ and of ratios $|A_j|/A_j$. In these terms, however the behaviour is asymptotically determined by the highest Λ-terms, so the conclusion remains even if we drop the lower terms.

By assumption, for all k, the sequence $(r+mN)\theta_k$ is uniformly distributed modulo 1. It follows that the values $e^{2i\pi(r+mN)\theta_k}$ are dense in the unit circle. If for some m, $g_1(r+mN) < 0$, then $g_1(r+Nm) \le -\varepsilon$ for some $\epsilon > 0$. Then, because of the density argument, there are arbitrarily large values of i for which $g_1(r+m_iN) \le 0$ contradicting condition 2 of the statement. Hence $g_1(r+mN) \ge 0$ for each m large enough. As g_1 is not a constant, there must be some m_0 so that $g_1(m_0) \ge \epsilon > 0$.

[6] This is the only point we need the assumption that the matrix entries are algebraic.

Next, let $R(x_1, \ldots, x_s)$ be a function obtained from g_1 by replacing each occurrence of $e^{i\theta_k n}$ by a variable x_k, hence each x_k will assume its value in the unit circle. Moreover, by the assumptions of the theorem, the values of x_k will be uniformly distributed in the unit circle.

Note that $g_1(n) = R((e^{2i\pi(r+m_iN)\theta_k})_{k\in A})$. Then, because the sequences $((r + m_iN)\theta_k)_i$ 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 $g_1(r + m_iM)$ with arbitrary precision. The function R is continuous, therefore there exists an interval $I = (x_1, y_1, ...) = ((x_k, y_k))_{k\in A}$ on which $R((x_k)) > \frac{\varepsilon}{2}$. So, if m_i is large enough and satisfies

$$((r + m_iN)\theta_1 \mod 1, \ldots) = ((r + m_iM)\theta_k \mod 1)_{k\in A} \in I,$$

then $g_1(r + m_iN) > \frac{\varepsilon}{2}$, which implies $f_A(r + m_iN) > 0$ and hence $a^{r+m_iN} \in L$. Now we just have to prove that the sequence $(r + m_iN)$ is "dense enough" to have $\underline{dens}(L) > 0$, contradicting again condition 1.
Then, because of uniform distribution imposed by condition 2, one has

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

And so for i large enough, $\frac{C(I,r+m_iN)}{r+m_iN} \geq \frac{d}{2}$, with $a^{h+n_iQ} \in L$, implying $\underline{dens}(L) > 0$, a contradiction. □

Corollary 1. *Let P be any polynomial with nonnegative coefficients and* $\deg(P) > 2$. *The language* $\{a^{P(n)} \mid n \in \mathbb{N}\}$ *is not in* $\mathsf{AfL}_{\mathbb{A}}$.

Corollary 2. *The language* $\{a^p \mid p \text{ prime}\}$ *is not in* $\mathsf{AfL}_{\mathbb{A}}$.

Proof (Proof of Corollary 1 and Corollary 2.). Turakainen proved that these two languages satisfies the two conditions of Theorem 6 [20]. Therefore, these two languages not in $\mathsf{AfL}_{\mathbb{A}}$. □

Acknowledgments

Yakaryılmaz was partially supported by Akadēmiskā personāla atjaunotne un kompetenču pilnveide Latvijas Universitātē līg Nr. 8.2.2.0/18/A/010 LU reģistrācijas Nr. ESS2018/289 and ERC Advanced Grant MQC. Hirvensalo was partially supported by the Väisälä Foundation and Moutot by ANR project CoCoGro (ANR-16-CE40-0005).

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 Yakaryılmaz. Automata and Quantum Computing. *CoRR*, abs/1507.0:1–32, 2015.
4. Aleksandrs Belovs, Juan Andrés Montoya, and Abuzer Yakaryılmaz. Can one quantum bit separate any pair of words with zero-error? *Tech. Rep.*, 1602.07967, arXiv, 2016.
5. Alejandro Díaz-Caro and Abuzer Yakaryılmaz. 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. J.-H. Evertse. On sums of S-units and linear recurrences. Compositio Math., 53(2):225–244,1984.
7. Georges Hansel. A simple proof of the skolem-mahler-lech theorem. Theoretical Computer Science, 43(1):91–98, 1986.
8. Mika Hirvensalo, Etienne Moutot, and Abuzer Yakaryılmaz: On the computational power of affine automata. Lecture Notes in Computer Science 10168 (Proceedings of LATA 2017), pp. 405–417, 2017.
9. Rishat Ibrahimov, Kamil Khadiev, Krišjānis Prūsis, Abuzer Yakaryılmaz: Error-Free Affine, Unitary, and Probabilistic OBDDs International Conference on Descriptional Complexity of Formal Systems, pp. 175–187, 2018
10. Emmanuel Jeandel. Topological automata. *Theory of Computing Systems*, 40(4):397–407, 2007.
11. Attila Kondacs and John Watrous. On the power of quantum finite state automata In FOCS, pages 66-75. IEEE, 1997
12. Ioan I Macarie. Space-Efficient Deterministic Simulation of Probabilistic Automata. *SIAM Journal on Computing*, 27(2):448–465, 1998.
13. Abuzer Yakaryılmaz and A. C. Cem Say. Languages recognized by nondeterministic quantum finite automata. Quantum Information & Computation, 10(9&10): 747-770, 2010
14. Azaria Paz. *Introduction to Probabilistic Automata (Computer Science and Applied Mathematics)*. Academic Press, Inc., Orlando, FL, USA, 1971.
15. M. O. Rabin. Probabilistic automata. Information and Control, 6:230-243, 1963.
16. Michael Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1st edition, 1996.
17. Paavo Turakainen: *On Probabilistic Automata and their Generalizations*. Annales Academiae Scientiarum Fennicae. Series A 429 (1969).
18. Paavo Turakainen: *On Languages Representable in Rational Probabilistic Automata*. Annales Academiae Scientiarum Fennicae. Series A 439 (1969).
19. Paavo Turakainen. Generalized Automata and Stochastic Languages. *Proceedings of the American Mathematical Society*, 21(2):303–309, 1969.
20. Paavo Turakainen. On nonstochastic languages and homomorphic images of stochastic languages. *Information Sciences*, 24(3):229–253, aug 1981.
21. Marcos Villagra and Abuzer Yakaryilmaz. *Language Recognition Power and Succinctness of Affine Automata*, pages 116–129. Springer International Publishing, Cham, 2016.