Natural Computing (2021) 20:259–270
https://doi.org/10.1007/s11047-020-09815-1

Computational limitations of affine automata and generalized affine automata

Mika Hirvensalo[1,2] · Etienne Moutot[3] · Abuzer Yakaryılmaz[1,4]

Accepted: 12 October 2020 / Published online: 18 January 2021

Abstract

We present new results on the computational limitations of affine automata (AfAs). First, we show that using the endmarker does not increase the computational power of AfAs. Second, we show that the computation of bounded-error rational-valued AfAs can be simulated in logarithmic space. Third, we identify some logspace unary languages that are not recognized by algebraic-valued AfAs. Fourth, we show that using arbitrary real-valued transition matrices and state vectors does not increase the computational power of AfAs in the unbounded-error model. When focusing only the rational values, we obtain the same result also for bounded error. As a consequence, we show that the class of bounded-error affine languages remains the same when the AfAs are restricted to use rational numbers only.

Keywords Non-classical models of automata · Affine automata · Logarithmic space · Generalized automata · Cutpoint languages · Bounded error

1 Introduction

Finite automata are interesting models to study since they express a very natural limitation of finite memory. They are also an interesting starting point for many computational models, since they are simpler than many others like pushdown automata or Turing machines. Due to this simplicity, there exist many different models of finite automata, all trying to express different computational settings. Deterministic (Sipser 2013), probabilistic (Paz 1971) and quantum (Ambainis and Yakaryılmaz 2015) 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 computational concept, called *affine computation* (Díaz-Caro and Yakaryılmaz 2016). 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 (Kondacs and Watrous 1997) or Latvian (Ambainis et al. 2006) quantum automata models.

AfAs and their certain generalizations have been investigated in a series of works by Díaz-Caro and Yakaryılmaz (2016), Villagra and Yakaryılmaz (2018), Belovs et al. (2017), Hirvensalo et al. (2017), Nakanish et al. (2017), Ibrahimov et al. (2018). In most of the cases, affine models (e.g., bounded-error and unbouded-error

A preliminary version appeared as "Hirvensalo, M., Moutot, E., Yakaryılmaz, A.: Computational Limitations of Affine Automata. In: Unconventional Computation and Natural Computations. LNCS, vol. 11493, pp. 108–121. Springer (2019)" (Hirvensalo et al. 2019). The arXiv number is 1904.02428.

✉ Mika Hirvensalo
mikhirve@utu.fi

Etienne Moutot
etienne.moutot@ens-lyon.org

Abuzer Yakaryılmaz
abuzer@lu.lv

[1] Department of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland

[2] Turku Centre for Computer Science (TUCS), Turku, Finland

[3] Aix-Marseille Univ., Toulon Univ., CNRS, LIS, Marseille, France

[4] Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Rīga, Latvia

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 new results. First, we show that using end-marker does not increase the computational power of affine automata with unbounded error or bounded error. Second, we show that the computation of bounded-error rational-valued affine automata is simulated in logarithmic space, and so we answer positively one of the open problems in Díaz-Caro and Yakaryılmaz (2016). Third, 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 cutpoints, improving a previous result showing that the same languages cannot be recognized with bounded error (Hirvensalo et al. 2017).

Moreover, we give the formal definition of generalized AfAs by allowing to use arbitrary real-valued transition matrices and state vectors. Fourth, we show that such generalization does not increase the computational power of AfAs with cutpoint language recognition. If we restricted these generalized AfAs to use only rational numbers, we obtain the same result also for bounded error language recognition. As a consequence, we show that the class of bounded-error affine languages remains the same when the AfAs are restricted to use rational numbers or only integers.

We provide all definitions in the next section and our results regarding using end-marker in Sect. 2.4. Our logarithmic space simulation is given in Sect. 3. Our impossibility result is given in Sect. 4. Our results related to generalized AfAs are given in Sect. 5.

A preliminary version of this paper was presented in UCNC2019 (Hirvensalo et al. 2019). In this version, Sects. 2.4 and 5 are completely new.

2 Preliminaries

Throughout the paper, Σ denotes the input alphabet – not containing letter \$ (we fix it as the right end-marker wherever it is used) , and $\widetilde{\Sigma} = \Sigma \cup \{\$\}$. The empty word is represented as ε. The set of all words defined on the alphabet Σ is denoted Σ^*. For any given word $w \in \Sigma^*$, $|w|$ is the length of w, we define $\tilde{w} = w\$$, and, if $w \neq \varepsilon$, w_i represents its i-th letter, where $1 \leq i \leq |w|$.

For any given class C, $\mathsf{C}_{\mathbb{Q}}$ and $\mathsf{C}_{\mathbb{A}}$ denote 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 basic notions 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.

A k-state probabilistic finite automaton (PFA) P over alphabet Σ is a triplet $P = (\mathbf{x}, \{M_i \mid i \in \Sigma\}, \mathbf{y})$ where $\mathbf{x} \in \mathbb{R}^k$ is a stochastic vector called *initial distribution*, each $M_i \in \mathbb{R}^{k \times k}$ is a stochastic matrix, and $\mathbf{y} \in \{0,1\}^k$ is the final vector (each 1 in $\mathbf{y}$ represents an accepting state).

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

$$f_P(w) = \mathbf{y}^T M_w \mathbf{x}. \tag{1}$$

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 usual ones, see Turakainen (1969)), but affine automata introduce also a non-linear behaviour. The automaton acts like a usual generalized probabilistic automaton until the last operation, which is a non-linear operation called a *weighting operation*.

A vector $\mathbf{v} \in \mathbb{R}^k$ is an affine vector if and only if its coordinates sum up to 1. A matrix M is an affine matrix if and only if all its columns are affine vectors. It is easy to verify that the multiplication of an affine matrix with an affine vector is also an affine vector, which ensures that affine automata are well defined.

A k-state *AfA* A over alphabet Σ is a triplet $A = (\mathbf{x}, \{M_i | i \in \Sigma\}, F)$, where $\mathbf{x}$ is an initial affine vector, each M_i is an affine transition matrix, and $F = \operatorname{diag}(\delta_1, \ldots, \delta_n)$ is the final projection matrix, where each $\delta_i \in \{0,1\}$.

The value computed by an affine automaton can be defined most conveniently via the following notation: $|\mathbf{v}| = \sum_i |v_i|$ stands for the usual L^1 norm. The final value of the affine automaton A is

$$f_A(w) = \frac{|FM_w\mathbf{x}|}{|M_w\mathbf{x}|}. \tag{2}$$

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

Remark that the final value for PFAs (1) is defined as matrix product $\mathbf{v}_f \mapsto \mathbf{y}^T\mathbf{v}_f$, which is a linear operation on $\mathbf{v}_f$.

On the other hand, computing final value from $\mathbf{v}_f$ as in (2) involves nonlinear operations $\mathbf{v}_f \mapsto \frac{|F\mathbf{v}_f|}{|\mathbf{v}_f|}$ such as L^1-norm and normalization (division).

2.2 Language recognition

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

A language $L \subseteq \Sigma^*$ is recognized by an automaton A with cutpoint $\lambda \in [0, 1)$ if and only if $L = \{w \in \Sigma^* \,|\, f_A(w) > \lambda\}$. These languages are called cutpoint languages.

A language $L \subseteq \Sigma^*$ is recognized by an automaton A with exclusive cutpoint $\lambda \in [0, 1]$ if and only if $L = \{w \in \Sigma^* \,|\, f_A(w) \neq \lambda\}$. These languages are called exclusive cutpoint languages.

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 $\forall w \notin L, f_A(w) \leq \lambda - \delta$. By fixing $\lambda = \frac{1}{2}$, we define language recognition with bounded error: A language L is recognized by an automaton A with bound error if and only if there exists an error bound $\epsilon \in [0, 1/2)$ such that $\forall w \in L,\ f_A(w) \geq 1 - \epsilon$ and $\forall w \notin L,\ f_A(w) \leq \epsilon$.

It is known that if a language recognized by a AfA (or PFA) with bounded error, then the error bound can be arbitrarily close to 0 (Hirvensalo et al. 2017).

2.3 Language classes

In the case of probabilistic (resp., affine automata), the set of cutpoint 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 (Paz 1971; Díaz-Caro and Yakaryılmaz 2016).

In the case of probabilistic (resp., affine automata), the set of exclusive cutpoint languages are called *exclusive stochastic languages* (resp., *exclusive affine languages*) and denoted by $\mathsf{SL}^{\neq}$ (resp., $\mathsf{AfL}^{\neq}$). The complements of the languages in $\mathsf{SL}^{\neq}$ (resp., $\mathsf{AfL}^{\neq}$) form $\mathsf{SL}^{=}$ (resp., $\mathsf{AfL}^{=}$). (Fixing the cutpoint in the interval (0, 1) does not change the classes $\mathsf{SL}^{\neq}$, $\mathsf{SL}^{=}$, $\mathsf{AfL}^{\neq}$, and $\mathsf{AfL}^{=}$ (Paz 1971; Yakaryılmaz and Say 2010; Díaz-Caro and Yakaryılmaz 2016).

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

A classical result by Rabin (1963) 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 (Ambainis and Yakaryılmaz 2015): 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 (Jeandel 2007). Affine automata do not have these properties, and in fact, they can recognize more than regular languages with bounded error (Díaz-Caro and Yakaryılmaz 2016).

2.4 Models using the right end-marker

A PFA or AfA can be defined by reading an extra letter ($M_\$$) for post-processing after reading the whole input. That is, the automaton reads $\tilde{w} = w\$$ for a given input word $w \in \Sigma^*$. Any such AfA (the definition of any such PFA is similar) can be formally defined as $A = (\mathbf{x}, \{M_i \mid i \in \widetilde{\Sigma}\}, F)$, and the accepting probability of the input w is calculated as $f_A(w) = \frac{|FM_{\tilde{w}}\mathbf{x}|}{|M_{\tilde{w}}\mathbf{x}|}$. Moreover, $\mathbf{v}_f = M_{\tilde{w}}\mathbf{x} = M_\$ M_w\mathbf{x}$.

It is known that, for any k-state PFA using the right end-marker, there is an equivalent k^2-state PFA without using the right end-marker such that, for any input word, both automata have the same accepting probabilities (Turakainen 1969). Even though we do not know whether this result is valid for AfAs or not, we can still show that post-processing does not increase the computational power of AfAs in the case of recognition with cutpoint or bounded error.

Theorem 1 *For a given* k*-state AfA* $A = (\mathbf{x}, \{M_i \mid i \in \widetilde{\Sigma}\}, F)$ *using the end-marker and for a given cutpoint* $\lambda \in [0, 1]$, *there is a* $4k$*-state AfA* $A' = (\mathbf{x}', \{M'_i \mid i \in \Sigma\}, F')$ *not using the end-marker such that, for any* $w \in \Sigma^*$, *both of* $f_A(w)$ *and* $f_{A'}(w)$ *are greater than* λ *or equal to* λ *or less than* λ.

Proof Let $w \in \Sigma^*$ be the given input of length $n \geq 0$. Let $\mathbf{v}_0 = \mathbf{x}$ and $\mathbf{u}_0 = M_\$\mathbf{v}_0$, and similarly, whenever $n > 0$, let $\mathbf{v}_l = M_{w_l}M_{w_{l-1}}\cdots M_{w_1}\mathbf{x}$ and $\mathbf{u}_l = M_\$\mathbf{v}_l$, where $1 \leq l \leq n$. Remark that $\mathbf{v}_f = \mathbf{u}_n$.

We define

$$\mathbf{v}'_0 = \mathbf{x}'_0 = \begin{pmatrix} \lambda\mathbf{v}_0 \\ (1-\lambda)\mathbf{v}_0 \\ \mathbf{u}_0 \\ -\mathbf{u}_0 \end{pmatrix}.$$

It is clear that the summation of entries are 1 and so $\mathbf{v}'_0$ is an affine state. For any $i \in \Sigma$, M'_i is defined as

$$\begin{pmatrix} M_i & 0 & I & I \\ 0 & M_i & 0 & 0 \\ M_{\$}M_i & M_{\$}M_i & 0 & 0 \\ -M_{\$}M_i & -M_{\$}M_i & 0 & 0 \end{pmatrix}.$$

It is easy to see that the entry summation of each column of M'_i is equal to 1, and so M'_i is an affine transition matrix. The multiplication of transition matrices with state vectors is trivial, and so we can easily obtain that

$$\mathbf{v}'_f = \mathbf{v}'_n = \begin{pmatrix} \lambda\mathbf{v}_n \\ (1-\lambda)\mathbf{v}_n \\ \mathbf{u}_n \\ -\mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \lambda\mathbf{v}_n \\ (1-\lambda)\mathbf{v}_n \\ \mathbf{v}_f \\ -\mathbf{v}_f \end{pmatrix}.$$

Let $f_A(w) = \frac{|F\mathbf{v}_f|}{|\mathbf{v}_f|} = \lambda + d$ for some real number d. We can derive that $|F\mathbf{v}_f| = |\mathbf{v}_f|(\lambda + d)$. We define F' as

$$\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & F & 0 \\ 0 & 0 & 0 & F \end{pmatrix}.$$

Then, we can calculate $f_{A'}(w)$ as follows:

$$\begin{aligned} f_{A'}(w) =& \frac{|\lambda\mathbf{v_n}| + |F\mathbf{v}_f| + |-F\mathbf{v}_f|}{|\lambda\mathbf{v}_n| + |(1-\lambda)\mathbf{v}_n| + 2|\mathbf{v}_f|} = \frac{\lambda|\mathbf{v}_n| + 2\lambda|\mathbf{v}_f| + 2d|\mathbf{v}_f|}{|\mathbf{v}_n| + 2|\mathbf{v}_f|} \\ =& \lambda + d\left(\frac{2|\mathbf{v}_f|}{|\mathbf{v}_n| + 2|\mathbf{v}_f|}\right) = \lambda + d', \end{aligned}$$

where either $d = d' = 0$ or both d and d' have the same sign. □

Corollary 1 *Any language recognized by an AfA using the right end-marker with a cutpoint (or an exclusive cutpoint) can be recognized by another AfA not using the right end-marker with the same cutpoint.*

Theorem 2 *Any language L recognized by a k-state AfA $A = (\mathbf{x}, \{M_i \mid i \in \widetilde{\Sigma}\}, F)$ using the right end-marker with error bound $\frac{1}{10}$ can be recognized by a $3k$-state AfA $A' = (\mathbf{x}', \{M_i \mid i \in \Sigma\}, F')$ not using the right end-marker with error bound $\frac{2}{10}$.*

Proof We use the same terminology in the previous proof. Let $m' = |\mathbf{x}|$ and let $m > 1$ be a real number satisfying $|M_i v| < m|v|$ for any $i \in \Sigma$ and for any affine vector v.

Let $w \in \Sigma^*$ be an input of length $n \geq 0$. We define $\mathbf{x}' = \begin{pmatrix} \mathbf{x} \\ 5mm'M_{\$}\mathbf{x} \\ -5mm'M_{\$}\mathbf{x} \end{pmatrix}$ and $M'_i = \left(\begin{array}{r|c|c} mM_i & I & I \\ \hline 5m'mM_{\$}M_i & 0 & 0 \\ \hline -5m'mM_{\$}M_i & 0 & 0 \end{array}\right)$ for any $i \in \Sigma$. Then, we obtain $\mathbf{v}'_f = \begin{pmatrix} m^n\mathbf{v}_n \\ 5m'm^n\mathbf{v}_f \\ -5m'm^n\mathbf{v}_f \end{pmatrix}$. We define $F' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & F \end{pmatrix}$. We know that $|\mathbf{v}_n| < m'm^n$. The accepting probability of A' on w is

$$f_{A'}(w) = \frac{|F'\mathbf{v}'_f|}{|\mathbf{v}'_f|} = \frac{10m'm^n|F\mathbf{v}_f|}{|\mathbf{v}_n| + 10m'm^n|\mathbf{v}_f|}.$$

If $w \in L$, then $10|F\mathbf{v}_f| \geq 9|\mathbf{v}_f|$ and

$$f_{A'}(w) \geq \frac{9m'm^n|\mathbf{v}_f|}{|\mathbf{v}_n| + 10m'm^n|\mathbf{v}_f|} > \frac{9m'm^n|\mathbf{v}_f|}{11m'm^n|\mathbf{v}_f|} > 0.8181.$$

If $w \notin L$, then $10|F\mathbf{v}_f| \leq |\mathbf{v}_f|$ and

$$f_{A'}(w) \leq \frac{m'm^n|\mathbf{v}_f|}{|\mathbf{v}_n| + 10m'm^n|\mathbf{v}_f|} < \frac{m'm^n|\mathbf{v}_f|}{10m'm^n|\mathbf{v}_f|} = 0.1$$

Therefore, L is recognized by A' with error bound $\frac{2}{10}$. □

3 Logarithmic simulation

Macarie (1998) 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 1 Notation $(b \bmod c)$ stands for the least nonnegative integer a satisfying $a \equiv b\,(\mathrm{mod} c)$. If $\mathbf{x} = (x_1, \ldots, x_r)$ and $\mathbf{n} = (n_1, \ldots, n_r) \in \mathbb{Z}^r$, we define $\mathbf{x}\,(\mathrm{mod}\ \mathbf{n}) = ((x_1 \bmod n_1), \ldots, (x_r \bmod n_r))$. Analogously, for any matrix $A \in \mathbb{Z}^{k\times k}$, we define $(A(\mathrm{mod\ n}))_{ij} = (A_{ij} \bmod n)$.

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

$$x \equiv a_1(\mathrm{mod\ n_1}), \ldots, x \equiv a_r(\mathrm{mod\ n_r}), \tag{3}$$

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

Remark 1 The Chinese Remainder Theorem implies 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) $\mathbf{n} = (n_1, \ldots, n_r)$ consists of pairwise coprime integers and 2) the integers stay in interval of length $N - 1$, where $N = n_1 \cdot \cdots \cdot n_r$.

Remark 2 In order to ensure that $\mathbf{n} = (n_1, \ldots, n_r)$ 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 2 $\mathbf{p}_r$ is an r-tuple $\mathbf{p}_r = (3, 5, 7, \ldots, p_r)$ consisting of r first primes by excluding 2. For this selection, a consequence of the prime number theorem is that, asymptotically, $P_r = 3 \cdot 5 \cdot 7 \cdot \cdots \cdot p_r = \frac{1}{2} e^{(1+o(1)) r \ln r}$.

Definition 3 Let $\mathbf{p}_r$ be as before. Then for any integer x, the *residual representation* $\mathrm{Res}_{\mathbf{p}_r}(x)$ stands for an integer vector of the residues: $(x(\bmod 3), x(\bmod 5), x(\bmod 7), \ldots, x(\bmod \mathrm{p_r}))$.

Theorem 4 (*Macarie* 1998) $\mathsf{SL}_{\mathbb{Q}}^{=} \subseteq \mathsf{L}$.

Proof For a given alphabet Σ, let $L \subseteq \Sigma^*$ be a language in $\mathsf{SL}_{\mathbb{Q}}^{=}$ and $P = (\mathbf{x}, \{M_i \mid i \in \Sigma\}, \mathbf{y})$ 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

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

Since each $M_i \in \mathbb{Q}^{k \times k}$, there exists an integer D such that all entries of each matrix $M'_i = D M_i$ are integers, and (4) can be rewritten as

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

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 $M'_i = D M_i$ 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 $\mathbf{p}_r = (3, 5, 7, \ldots, p_r)$, we let $M_i^{(p)} = M'_i \bmod p$, and compute

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

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 $\mathbf{p}_r$. The comparison $2 f_{P'}(w) \equiv D^n \pmod{p}$ can be hence done in $O(\log p)$ space.

Reusing the space, the comparison can be made sequentially for each coordinate of $\mathbf{p}_r$, 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 $2 f_{P'}(w)$ 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 $2 f_{P'}(w) = D^n$. Since $f_{P'}(w) \leq D^n$, we need to select r so that $P_r > 2D^n$, 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* 1998) *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

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

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 $\mathrm{Res}_{\mathbf{p}_r}(x)$ and $\mathrm{Res}_{\mathbf{p}_r}(y)$, and it is not straightforward how 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 Macarie* (1998)) *For any integer x and modulus $\mathbf{p}_r = (3, 5, 7, \ldots, p_r)$, there is a deterministic algorithm that given $\mathrm{Res}_{\mathbf{p}_r}(x)$ and $M \in \mathbb{Z}$ as input, produces the output $x(\bmod \mathrm{M})$ in space $O(\log p_r + \log M)$.*

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

Lemma 3 (***Claim modified from*** *Macarie (1998)*) *Let $\mathbf{p}_r = (3, 5, 7, \ldots, p_r)$ 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 of Theorem 4, testing the congruence sequentially for each prime. Testing $x \geq y$ is possible by Lemmata 1 and 2: First compute $\mathrm{Res}_{\mathbf{p}_r}(z) = \mathrm{Res}_{\mathbf{p}_r}(x) - \mathrm{Res}_{\mathbf{p}_r}(y) (\mathrm{mod}\ \mathbf{p}_r)$, 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 5 $\mathsf{SL}_{\mathbb{Q}} \subseteq \mathsf{L}$.

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, $2 \equiv -3 (\mathrm{mod}\ 5)$, but $|2| \not\equiv |-3| (\mathrm{mod}\ 5)$. 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 that introduced by Turakainen already in 1969 (Turakainen 1968, 1969).

Theorem 6 $\mathsf{AfL}_{\mathbb{Q}} \subseteq \mathsf{L}$.

Proof For a given alphabet Σ, let $L \in \Sigma^*$ be a language in $\mathsf{AfL}_{\mathbb{Q}}$ and $A = (\mathbf{x}, \{M_i \mid i \in \Sigma\}, F)$ 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

$$B_i = \begin{pmatrix} 0 & \mathbf{0}^T & 0 \\ \mathbf{c}_i & M_i & \mathbf{0} \\ e_i & \mathbf{d}_i^T & 0 \end{pmatrix},$$

where $\mathbf{c}_i$, $\mathbf{d}_i$, and e_i are chosen so that the column and row sums of B_i are zero. We define $\mathbf{x}' = \begin{pmatrix} 0 \\ \mathbf{x} \\ 0 \end{pmatrix}$ 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 $|B_w \mathbf{x}'| = |M_w \mathbf{x}|$ as well as $|F' B_w \mathbf{x}'| = |F M_w \mathbf{x}|$.

For the next step, we introduce an $(k+2) \times (k+2)$ 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 non-negativity of the matrix entries of each C_i. It follows that

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

and

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

Similarly,

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

Now

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

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

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

Hence the inequality

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

is equivalent to

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

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

For that end, the residue representation of vector $\mathbf{a} = F'D_w\mathbf{x}' \in \mathbb{R}^{k+2}$ can be obtained in logarithmic space as in the proof of Theorem 4.

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

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

it is sufficient to decide whether $\mathbf{a}_i \geq \mathbf{b}_i$ holds. As the residue representations for each $\mathbf{a}_i$ and $\mathbf{b}_i$ 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.[1] 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 (Hirvensalo et al. 2017), we applied the method of Turakainen (1981) 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 $\mathsf{AfL}_{\mathbb{A}}$.

Definition 4 (**Lower density**) Let $L \subseteq a^*$ be a unary language. We call **lower density** of L the limit

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

Definition 5 (**Uniformly distributed sequence**) Let $(\mathbf{x}_n)$ 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 $C(I,n) = |\{\mathbf{x}_i \bmod 1 \in I | 1 \leq i \leq n\}|$.

We say that $(\mathbf{x}_n)$ 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 7 *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 $a^{r+m_iN} \subseteq L$ and for any irrational number α, the sequence $((r + m_iN)\alpha)$ 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 $\mathbf{v}$ such that the acceptance value of A is

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

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 $M^n\mathbf{v}$ have the form

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

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

$$\frac{|F(\alpha M)^n\mathbf{v}|}{|(\alpha M)^n\mathbf{v}|} = \frac{|\alpha^n FM^n\mathbf{v}|}{|\alpha^n M^n\mathbf{v}|} = \frac{|FM^n\mathbf{v}|}{|M^n\mathbf{v}|}.$$

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.[2]

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

[2] Note that the new matrix obtained may not be affine, so it would be wrong to assume that all AfAs have 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 $\frac{|P(\alpha M)^n\mathbf{v}|}{|(\alpha M)^n\mathbf{v}|} = \frac{|PM^n\mathbf{v}|}{|M^n\mathbf{v}|}$ take.

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

$$N = \operatorname{lcm}\{\operatorname{ord}(\lambda_i/\lambda_j) \mid i \neq j \text{ and } \lambda_i/\lambda_j \text{ is a root of unity}\}, \tag{11}$$

equation (10) becomes

$$(M^{r+mN}\mathbf{v})_j = \sum_{k=1}^{s} p_{jk}(r+mN)\lambda_k^r(\lambda_k)^{Nm} = \sum_{k=1}^{s'} q_{jk}(m)\mu_k^m, \tag{12}$$

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

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

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), $g(n) := \sum_{j\in E_a}\left|(M^n\mathbf{v})_j\right| - \sum_{j\in\overline{E_a}}\left|(M^n\mathbf{v})_j\right|$ 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 $\lim_{n\to\infty}\frac{t_2(n)}{t_1(n)} = 0$.

We can organize the terms in expression (10) as

$$(M^n\mathbf{v})_j = \sum_{k=1}^{s} p_{jk}(n)\lambda_k^n = \Lambda_j^{(N)}(n) + \Lambda_j^{(N-1)}(n) + \cdots + \Lambda_j^{(0)}(n), \tag{13}$$

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

$$\Lambda_j^{(m)}(n) = \sum_{k=1}^{m_j} c_{mk} n^{d_m}\lambda_{mk}{}^n = n^{d_m}\lambda_m^n\sum_{k=1}^{m_j} c_{mk}e^{2\pi i n\theta_{mk}} \tag{14}$$

(for notational simplicity, we mostly omit the dependency on j in the right hand side of (14)). 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 (13) 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

$$\begin{aligned} g(n) &= \sum_{j\in E_a}\left|\sum_{k=1}^{s} p_{jk}(n)\lambda_k^n\right| - \sum_{j\in\overline{E_a}}\left|\sum_{k=1}^{s} p_{jk}(n)\lambda_k^n\right| \\ &= \sum_{j\in E_a}\left|A_j(n) + B_j(n) + C_j(n)\right| \\ &\quad - \sum_{j\in\overline{E_a}}\left|A_j(n) + B_j(n) + C_j(n)\right|, \end{aligned} \tag{15}$$

where $A_j(n) + B_j(n) + C_j(n)$ is a grouping of all Λ-terms in (13) 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

$$A = \sum_{j\in E_a}\left|A_j(n)\right| - \sum_{j\in\overline{E_a}}\left|A_j(n)\right| \tag{16}$$

   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 (16) 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 $A_j(n)$, and
3. $C_j(n)$ contains the rest of the Λ-terms, lower than $B_j(n)$

□

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

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

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

Now

$$\begin{aligned} |A+z| &= |A|\left|1+\frac{z}{A}\right| = |A|\left(1+\mathrm{Re}\frac{z}{A}+O\left(\left(\frac{z}{A}\right)^2\right)\right) \\ &= |A| + \mathrm{Re}\frac{|A|}{A}z + O\left(\frac{z^2}{A}\right). \end{aligned}$$

◁

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 (15) 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 (Hansel 1986), 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 Evertse (1984), then $|A_j(n)| = \Omega(n^d\lambda^{n-\epsilon})$ for each $\epsilon > 0$.[3] 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

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

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 $g_1(r + Nm) \leq -\varepsilon$ for some $\epsilon > 0$. Then, because of the density argument, there are arbitrarily large values of i for which $g_1(r + m_iN) \leq 0$ contradicting condition 2 of the statement. Hence $g_1(r + mN) \geq 0$ for each m large enough. As g_1 is not a constant, there must be some m_0 so that $g_1(m_0) \geq \epsilon > 0$.

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 $I = (x_1, y_1, \ldots) = ((x_k, y_k))_{k \in A}$ on which $R((x_k)) > \frac{\varepsilon}{2}$. So, if m_i is large enough and satisfies

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

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

Corollary 3 *The language* $\{a^p | p \text{ is prime}\}$ *is not in* $\mathsf{AfL}_{\mathbb{A}}$.

Proof of Corollary 2 and Corollary 3 Turakainen proved that these two languages satisfies the two conditions of Theorem 7 (Turakainen 1981). Therefore, these two languages not in $\mathsf{AfL}_{\mathbb{A}}$. □

5 Generalized affine automata

In this section, we show that using arbitrary real state vector and transition matrices does not increase the computational power of AfAs. A generalized affine finite automaton (GAfA) is a 3-tuple $G = (\mathbf{x}, \{M_i | i \in \widetilde{\Sigma}\}, F)$, where, different from an AfA, $\{M_i | i \in \widetilde{\Sigma}\}$ is the set of real-valued transition matrices without any restriction on the

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

column summations and $\mathbf{x}$ is the real-valued initial state vector. The final affine state of G on the given input $w \in \Sigma^*$ for some $n \geq 0$ is

$$\mathbf{v}_f = M_{\$}M_w\mathbf{x} = M_{\$}M_{w_n} \cdots M_{w_1}\mathbf{x},$$

where $M_\varepsilon = I$. It must be guaranteed that at least one entry of $\mathbf{v}_f$ is non-zero for any possible input. The *accepting probability* of G on w is calculated in the same way of an AfA: $f_G(w) = \frac{|F\mathbf{v}_f|}{|\mathbf{v}_f|}$.

We start with proving that GAfAs with cutpoint define the same class of languages as AfAs with cutpoint.

Theorem 8 *Any language L recognized by a k-state GAfA* $G = (\mathbf{x}, \{M_i | i \in \widetilde{\Sigma}\}, F)$ *with cutpoint* $\lambda \in [0, 1)$ *is recognized by a* $(k+2)$*-state AfA* $A = (\mathbf{x}', \{M_i | i \in \widetilde{\Sigma}\}, F')$ *with cutpoint* λ.

Proof Let $t_0 = 1 - \sum_{i=1}^{k} \mathbf{x}_i$. We define $\mathbf{x}' = \begin{pmatrix} \mathbf{x} \\ \lambda t_0 \\ (1-\lambda)t_0 \end{pmatrix}$.

For letter $i \in \Sigma$, let c_j be the j-th column summation of M_i and $d_j = 1 - c_j$. We define M_i' based on M_i:

$$\left(\begin{array}{cccc|c} & M_i & & & \mathbf{0} \\ \hline \lambda d_1 & \lambda d_2 & \cdots & \lambda d_k & 1\ 0 \\ (1-\lambda)d_1 & (1-\lambda)d_2 & \cdots & (1-\lambda)d_k & 0\ 1 \end{array}\right),$$

where each column summation is 1. Then, we can calculate $\mathbf{v}'_f$, for a given input $w \in \Sigma^*$, as $\begin{pmatrix} \mathbf{v}_f \\ \lambda t_f \\ (1-\lambda)t_f \end{pmatrix}$, where $t_f = 1 - \sum_{i=1}^{k} (\mathbf{v}_f)_i$. We define

$$F' = \begin{pmatrix} F & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Let $t = |t_f|$. We have $f_G(w) = \frac{|F\mathbf{v}_f|}{|\mathbf{v}_f|} = \lambda + d$ for some real number d. Then the accepting probability of w by A is

$$f_A(w) = \frac{|F'\mathbf{v}'_f|}{|\mathbf{v}'_f|} = \frac{|F\mathbf{v}_f| + \lambda t}{|\mathbf{v}_f| + t} = \frac{(\lambda + d)|\mathbf{v}_f| + \lambda t}{|\mathbf{v}_f| + t} = \lambda + \frac{d}{|\mathbf{v}_f| + t}.$$

Thus, both of $f_G(w)$ and $f_A(w)$ are greater than λ or equal to λ or less than λ. □

Remark that when the cutpoint is 0, then the constructed AfA can indeed use one state less in the above proof.

We can obtain the same result for bounded error case when focusing on the rational numbers. First we show that there is no difference between using rational numbers and integers.

Lemma 5 *For any given GAfA* $G_1 = (\mathbf{x}, \{M_i | i \in \widetilde{\Sigma}\}, F)$ *with rational number components, there is a GAfA* G_2 *with integer number components such that they have the same accepting probability on any input string.*

Proof Let z be sufficiently big integer such that zM_i for each $i \in \Sigma$ and $z\mathbf{x}$ contains only integers. Then, G_2 is defined as $(z\mathbf{x}, \{zM_i | i \in \widetilde{\Sigma}\}, F)$. Due to linearity, if the final vector of G_1 on a given input $w \in \Sigma^*$ is $\mathbf{v}_f$, then, the final vector of G_2 on a any given input is $z^{|\tilde{w}|+1}\mathbf{v}_f$. Thus, $f_{G_1}(w) = f_{G_2}(w)$. □

Theorem 9 *Any language L recognized by a k-state GAfA* $G = (\mathbf{x}, \{M_i | i \in \widetilde{\Sigma}\}, F)$ *with bounded error can be recognized by a* $(2k+1)$*-state AfA* $A = (\mathbf{x}', \{M_i' | i \in \widetilde{\Sigma}\}, F')$ *with bounded error, where both automata have only integer components.*

Proof Let $\frac{1}{2} - \frac{1}{m}$ for $m \geq 2$ be the error bound and $w \in \Sigma^*$ be the given input.

We define $\mathbf{x}' = \begin{pmatrix} \mathbf{x} \\ -\mathbf{x} \\ 1 \end{pmatrix}$. For each $i \in \Sigma$, we define $M_i' = \begin{pmatrix} M_i & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & M_i & \boldsymbol{0} \\ r_1 & r_1 & 1 \end{pmatrix}$, and for letter \$, we define $M_{\$}' = \begin{pmatrix} m^2 M_{\$} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & m^2 M_{\$} & \boldsymbol{0} \\ r_{\$} & r_{\$} & 1 \end{pmatrix}$, where r_i and $r_{\$}$ are row vectors guaranteeing that the entry summation of each corresponding column is 1. The final state vector of A on w can be easily obtained as $\mathbf{v}'_f = \begin{pmatrix} m^2\mathbf{v}_f \\ -m^2\mathbf{v}_f \\ 1 \end{pmatrix}$, where $\mathbf{v}_f$ is the final state vector of G on w.

We define $F' = \begin{pmatrix} F & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & 0 \end{pmatrix}$. Let $a = |F\mathbf{v}_f|$ and $r = |\mathbf{v}_f| - a$. Remark that a and r can be only non-negative integers. The accepting probability of A on w is

$$f_A(w) = \frac{2m^2 a}{2m^2 a + 2m^2 r + 1} \tag{17}$$

since we have two copies of $\mathbf{v}_f$ where one is multiplied by m^2 and the other is multiplied by $-m^2$. For any $w \notin L$, it is straightforward that

$$f_A(w) = \frac{2m^2 a}{2m^2 a + 2m^2 r + 1} \leq \frac{2m^2 a}{2m^2 a + 2m^2 r} = \frac{a}{a + r} = f_G(w).$$

In the remaining part, we focus on only the members: $w \in$

L and $\frac{a}{a+r} = \frac{1}{2} + c$ for some $\frac{1}{2} \geq c \geq \frac{1}{m}$. From the equation of $f_G(w)$, we can obtain $a + r = \frac{2a}{1+2c}$ and we can substitute $a + r$ with $\frac{2a}{1+2c}$ in equation (17):

$$f_A(w) = \frac{2m^2 a}{\frac{4m^2 a}{1+2c} + 1} = \frac{(1+2c)2m^2 a}{4m^2 a + 2c + 1}$$
$$= \frac{(1+2c)(2m^2 a + c + \frac{1}{2} - c - \frac{1}{2})}{4m^2 a + 2c + 1}.$$

After simplification, we have

$$f_A(w) = \frac{1}{2} + c - \frac{(2c+1)(c+\frac{1}{2})}{4m^2 a + 2c + 1} = \frac{1}{2} + c - \frac{(2c+1)^2}{8m^2 a + 4c + 2}.$$

We know that $a \geq 1$ ($a \neq 0$ for $w \in L$) and $c \leq \frac{1}{2}$. Thus, we can easily follow that

$$\frac{(2c+1)^2}{8m^2 a + 4c + 2} < \frac{4}{8m^2} = \frac{1}{2m^2}.$$

Hence, we can bound the accepting probability of any member from below as

$$f_A(w) > \frac{1}{2} + \frac{1}{m} - \frac{1}{2m^2} = \frac{1}{2} + \frac{2m-1}{2m^2}.$$

Since there is a constant gap for every member, we conclude that A recognizes L with bounded error. □

Villagra and Yakaryılmaz (2016), showed that one-sided error (either all members are accepted with probability 1 or all non-members are accepted with probability 0) versions of BAfL are the identical if they are defined by AfAs with rational number components or by AfAs with integer components. By using the above results, we can follow that the same result is valid also for (two-sided error class) BAfL.

Corollary 4 $\mathsf{BAfL}_{\mathbb{Q}} = \mathsf{BAfL}_{\mathbb{Z}}$.

Acknowledgements Open access funding provided by University of Turku (UTU) including Turku University Central Hospital. We thank to the anonymous reviewers for their helpful corrections and comments. M. Hirvensalo was partially supported by the Väisälä Foundation. E. Moutot was partially supported by ANR project CoCoGro (ANR-16-CE40-0005). A. Yakaryılmaz was partially supported by the ERDF project 1.1.1.5/18/A/020, Akadēmiskā personāla atjaunotne un kompetenču pilnveide Latvijas Universitātē līg Nr. 8.2.2.0/18/A/010 LU reģistrcijas Nr. ESS2018/289, and ERDF project Nr. 1.1.1.5/19/A/005 "Quantum computers with constant memory".

Funding Open access funding provided by University of Turku (UTU) including Turku University Central Hospital..

References

Ambainis A, Beaudry M, Golovkins M, Ķikusts A, Mercer M, Thérien D (2006) Algebraic results on quantum automata. Theory Comput Syst 39(1):165–188

Ambainis A, Yakaryılmaz A (2015) Automata: from mathematics to applications, chap. Automata and Quantum Computing (To appear). (arXiv:1507.01988)

Belovs A, Montoya JA, Yakaryılmaz A (2017) On a conjecture by Christian Choffrut. Int J Found Comput Sci 28(5):483–502

Díaz-Caro A, Yakaryılmaz A (2016) Affine computation and affine automaton. In: Computer science — Theory and applications, LNCS, vol 9691. Springer, pp 1–15. arXiv:1602.04732

Evertse JH (1984) On sums of S-units and linear recurrences. Compos Math 53(2):225–244

Hansel G (1986) Une démonstration simple du théorème de Skolem-Mahler-Lech. Theor Comput Sci 43:91–98

Hirvensalo M, Moutot E, Yakaryılmaz A (2017) On the computational power of affine automata. Language Autom Theory Appl LNCS 10168:405–417

Hirvensalo M, Moutot E, Yakaryılmaz A (2019) On the computational power of affine automata. Unconvent Comput Nat Comput, LNCS 11493:108–121

Ibrahimov R, Khadiev K, Prūsis K, Yakaryılmaz A (2018) Error-free affine, unitary, and probabilistic OBDDs. In: Descriptional complexity of formal systems, LNCS, vol 10952. Springer, pp 175–187

Jeandel E (2007) Topological automata. Theory Comput Syst 40(4):397–407

Kondacs A, Watrous J (1997) On the power of quantum finite state automata. In: FOCS'97, pp 66–75

Macarie II (1998) Space-efficient deterministic simulation of probabilistic automata. SIAM J Comput 27(2):448–465

Nakanish M, Khadiev K, Prūsis K, Vihrovs J, Yakaryılmaz A (2017) Exact affine counter automata. Electron Proc Theor Comp Sci EPTCS 252:205–218

Paz A (1971) Introduction to probabilistic automata. Academic Press, New York

Rabin MO (1963) Probabilistic automata. Inf Control 6:230–243

Sipser M (2013) Introduction to the Theory of Computation, 3rd edn. Cengage Learning, United States of America

Turakainen P (1968) On probabilistic automata and their generalizations. Annales Academiae Scientiarum Fennicae, Ser. A 429(1)

Turakainen P (1969) Generalized automata and stochastic languages. Proc Am Math Soc 21:303–309

Turakainen P (1981) On nonstochastic languages and homomorphic images of stochastic languages. Inf Sci 24(3):229–253

Villagra M, Yakaryılmaz A (2016) Language recognition power and succintness of affine automata. In: Unconventional computation and natural computation, LNCS, vol 9726. Springer, pp 116–129

Villagra M, Yakaryılmaz A (2018) Language recognition power and succinctness of affine automata. Natural Comput 17(2):283–293

Yakaryılmaz A, Say ACC (2010) Languages recognized by nondeterministic quantum finite automata. Quantum Inf Comput 10(9&10):747–770

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.