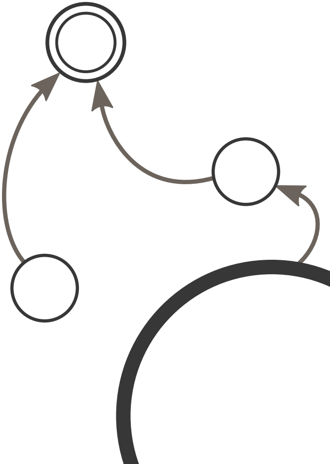
AFFINE AUTOMATA
power and limitations

STACS 2020
Montpellier, March 12, 2020



Mika Hirvensalo, **Etienne Moutot**
and Abuzer Yakaryılmaz

Mika Hirvensalo, Etienne Moutot and Abuzer Yakaryılmaz
»On the computational power of affine automata«
Language and Automata Theory and Applications (LATA) 2017

Mika Hirvensalo, Etienne Moutot and Abuzer Yakaryılmaz
»Computational Limitations of Affine Automata«
Unconventional Computation and Natural Computation (UCNC) 2019

DEFINITIONS AND MOTIVATION

DETERMINISTIC FINITE AUTOMATON

$$(\mathbf{M}_x)_{i,j} = 1 \text{ iff } j \xrightarrow{x} i$$

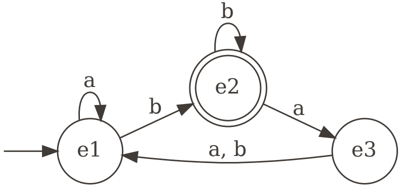


Figure: Example of Deterministic Finite Automaton

$$\mathbf{M}_a = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{M}_b = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

PROBABILISTIC FINITE AUTOMATON

$$(\mathbf{M}_x)_{i,j} = P(j \xrightarrow{x} i)$$

$\mathbf{M}_x$ are stochastic matrices

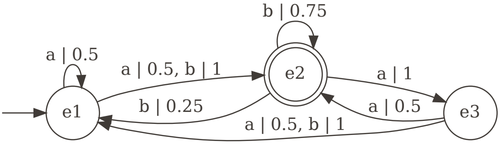


Figure: Example of Probabilistic Finite Automaton

$$\mathbf{M}_a = \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{M}_b = \begin{pmatrix} 0 & 0.25 & 1 \\ 1 & 0.75 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

PROBABILISTIC FINITE AUTOMATON

$w = x_1 \dots x_n$
$\mathbf{M}_w = \mathbf{M}_{x_n} \dots \mathbf{M}_{x_1}$
$\mathbf{P}$ projection associated to accepting states
$\mathbf{v}_0$ initial vector

$$f_P(w) = \mathbf{P}\mathbf{M}_w\mathbf{v}_0$$

STOCHASTIC LANGUAGES

Definition (Stochastic language - SL)

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

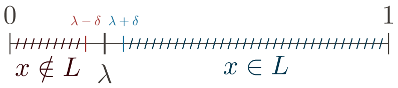
$\lambda \in (0, 1)$ is called the **cutpoint**

STOCHASTIC LANGUAGES

Definition (Isolated cutpoint - ISL)

$\lambda \in (0,1)$ is an **isolated cutpoint** iff there exist $\delta > 0$ such that

- → $\forall w \in L, f_P(w) \geq \lambda + \delta$
- → $\forall w \notin L, f_P(w) \leq \lambda - \delta$

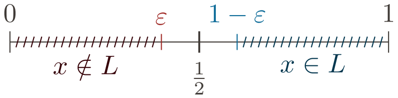


STOCHASTIC LANGUAGES

Definition (Bounded error - BSL)

L is recognized with **bounded error** iff there exists $\varepsilon \in (0, \frac{1}{2})$ such that

- → $\forall w \in L, f_P(w) \geq 1 - \varepsilon$
- → $\forall w \notin L, f_P(w) \leq \varepsilon$



STOCHASTIC LANGUAGES

Theorem

ISL = BSL = REG

GENERAL AUTOMATA ?

Theorem (Jeandel, 2007)

The bounded-error language of any topological automata with compact set of states and continuous evaluation function is REG.

AFFINE FINITE AUTOMATON

$$A = (E, \Sigma, \{\mathbf{M}_x \mid x \in \Sigma\}, \mathbf{v}_0, E_a)$$

- E finite set of states of P
- Σ finite alphabet
- $\{\mathbf{M}_x\}$ set of transition matrices, all columns sums up to 1
- $\mathbf{v}_0$ initial vector, coordinates sums up to 1
- E_a set of accepting states

AFFINE FINITE AUTOMATON

$w = x_1 \dots x_n$
$\mathbf{M}_w = \mathbf{M}_{x_n} \dots \mathbf{M}_{x_1}$
$\mathbf{P}$ projection associated to accepting states
$\mathbf{v}_0$ initial vector

$$f_A(w) = \frac{|\mathbf{P}\mathbf{M}_w\mathbf{v}_0|}{|\mathbf{M}_w\mathbf{v}_0|} = \frac{\sum_{e_i \in E_a} |(\mathbf{M}_w\mathbf{v}_0)_i|}{\sum_{e_i \in E} |(\mathbf{M}_w\mathbf{v}_0)_i|}$$

AFFINE LANGUAGE

Definition (Affine language - AfL)

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

$\lambda \in (0, 1)$ is called the **cutpoint**

BOUNDED ERROR

Definition (Isolated cutpoint - IAfL)

$\lambda \in (0,1)$ is an **isolated cutpoint** iff there exist $\delta > 0$ such that

- $\forall w \in L, f_P(w) \geq \lambda + \delta$
- $\forall w \notin L, f_P(w) \leq \lambda - \delta$

Definition (Bounded error - BAfL)

L is recognized with **bounded error** iff there exists $\varepsilon \in (0, \frac{1}{2})$ such that

- $\forall w \in L, f_P(w) \geq 1 - \varepsilon$
- $\forall w \notin L, f_P(w) \leq \varepsilon$

AFFINE LANGUAGES

Theorem (Díaz-Caro, Yakaryılmaz, 2016)

IAfL = BAfL

AFFINE LANGUAGES

Theorem (Díaz-Caro, Yakaryılmaz, 2016)

$\mathsf{IAfL} = \mathsf{BAfL} \neq \mathsf{REG}$

STABILITY RESULTS

OPERATION ON LANGUAGES

Theorem (Hirvensalo, M. & Yakaryılmaz, 2017)

BAfL is stable under **union**, **intersection** and **complement**.

PRODUCT AND SUM

Theorem

$f, g : \Sigma^* \to [0,1]$ *functions of affine automata.* fg *is affine too.*

Theorem

$f, g : \Sigma^* \to [0,1]$ *functions of affine automata.* $\alpha, \beta \geq 0$, $\alpha + \beta = 1$.

$\alpha f + \beta g$ *is affine too.*

OPERATIONS ON LANGUAGES ?

$L_A, L_B \in$ BAfL, with error bound ε

$L_+ = \{w \mid \frac{1}{2}(f_A(w) + f_B(w)) > \lambda\}$
$L_\times = \{w \mid f_A(w) \times f_B(w) > \lambda\}$

$$\varepsilon < \frac{1}{3} \Rightarrow L_+ \in \text{BAfL}$$

$$L_+ = L_A \cup L_B$$

$$\varepsilon < \frac{1}{3} \Rightarrow L_\times \in \text{BAfL}$$

$$L_\times = L_A \cap L_B$$

ERROR REDUCTION

Theorem (Error Reduction)

Let $L \in$ BAfL. There exists $\mathcal{A}$ such that:

- $\forall w \in L, f_{\mathcal{A}}(w) \geq \frac{3}{4}$
- $\forall w \notin L, f_{\mathcal{A}}(w) \leq \frac{1}{4}$

i.e. $\mathcal{A}$ recognizes L with error bound $\varepsilon = \frac{1}{4}$.

ERROR REDUCTION

Lemma

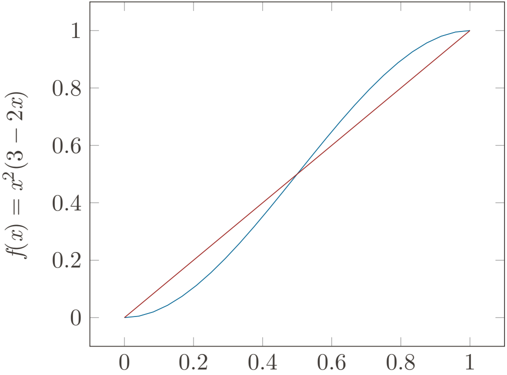
Let f be computed by an affine automaton $\mathcal{A}$. There exists an affine automaton $\mathcal{B}$ such that $f_{\mathcal{B}} = f^2(3-2f)$.

$$\mathbf{B}_x = \mathbf{A}_x \otimes \mathbf{A}_x \otimes \mathbf{A}_x$$

$$\mathbf{v}_0' = \mathbf{v}_0 \otimes \mathbf{v}_0 \otimes \mathbf{v}_0$$

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

ERROR REDUCTION



$$\varepsilon \longrightarrow \varepsilon^2(3 - 2\varepsilon) < \varepsilon$$

SPACE: BOUND AND SIMULATION

STATE VECTOR

Theorem (Hirvensalo, M. & Yakaryılmaz, 2017)

$L \in$ AfL can be recognized by an affine automaton with a **bounded state vector**, at any step of the computation.

STATE VECTOR

Theorem (Hirvensalo, M. & Yakaryılmaz, 2017)

$L \in$ AfL can be recognized by an affine automaton with a **bounded state vector**, at any step of the computation.

☺ We do not need unbounded space to recognize AfL

STATE VECTOR

Theorem (Hirvensalo, M. & Yakaryılmaz, 2017)

$L \in$ AfL can be recognized by an affine automaton with a **bounded state vector**, at any step of the computation.

- ☺ We do not need unbounded space to recognize AfL
- ☹ The proof does not preserve the error gap if there is one

STATE VECTOR (PROOF)

$\mathcal{A}$ initial (unbounded) automaton.
$C = \max_{x,i,j} |(A_x)_{i,j}|$

$$B_x = \left(\begin{array}{ccc|cc} & & & 0 & 0 \\ & \frac{1}{kC}A_x & & \vdots & \vdots \\ & & & 0 & 0 \\ \hline \frac{1}{2} - \frac{1}{2kC} & \cdots & \frac{1}{2} - \frac{1}{2kC} & 1 & 0 \\ \frac{1}{2} - \frac{1}{2kC} & \cdots & \frac{1}{2} - \frac{1}{2kC} & 0 & 1 \end{array}\right)$$

The new function is:

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

LOGARITHMIC SIMULATION (STOCHASTIC CASE)

Theorem (Macarie 1998)

$\mathsf{SL}_{\mathbb{Q}} \subseteq \mathsf{L}$ (Stochastic automata can be simulated in logspace)

LOGARITHMIC SIMULATION

Theorem (Hirvensalo, M. & Yakaryılmaz, 2019)

$\mathrm{AfL}_{\mathbb{Q}} \subseteq \mathrm{L}$ (Affine automata can be simulated in logspace)

MACARIE KEY IDEAS: MODULAR ARITHMETICS

Residue representation of $\mathbf{x}$ w.r.t $\mathbf{n}$:

$$\mathbf{x} \mod \mathbf{n} = (x_1 \mod n_1, \ldots, x_r \mod n_r)$$

Chinese Remainder Theorem:

1. $\mathbf{n} = (n_1, \ldots, n_r)$ consists of pairwise coprime integers
2. $\forall i, x_i \leq N - 1$ with $N = n_1 \cdots n_r$

Then $\mathbf{x}$ **can be recovered from its residue representation**

MACARIE KEY IDEAS: MODULAR ARITHMETICS

Complex operations on $\mathbf{x}$

$\rightarrow$ Do operations in $O(\log x)$ space

MACARIE KEY IDEAS: MODULAR ARITHMETICS

Complex operations on $\mathbf{x} \mod p$

$\rightarrow$ Do operations in $O(\log p)$ space

MACARIE KEY IDEAS: MODULAR ARITHMETICS

Complex operations on $\mathbf{x} \mod p$

$\rightarrow$ Do operations in $O(\log p)$ space

Prime number theorem:

$$P_r = 3 \cdot 5 \cdot 7 \cdot \ \cdots \ \cdot p_r = \frac{1}{2} e^{(1+o(1)) r \ln r}$$

$\rightarrow$ Do operations in $O(\log r)$ space on integers of size $\frac{1}{2} e^{(1+o(1)) r \ln r}$

AFFINE CASE: ABSOLUTE VALUES

Problem in the affine case: **the absolute values**

$2 \equiv -3 \mod 5$

But

$|2| \not\equiv |-3| \mod 5$

AFFINE CASE: ABSOLUTE VALUES

Problem in the affine case: **the absolute values**

$2 \equiv -3 \mod 5$

But

$|2| \not\equiv |-3| \mod 5$

Solution: finding a "close enough" automata with only non-negative entries

NON-AFFINE LANGUAGES

NON-BAFL LANGUAGES

P a polynomial with nonnegative coefficients and $\deg(P) > 2$,
POLY $= \{a^{P(n)} \mid n \in \mathbb{N}\}$.

PRIME $= \{a^p \mid p \text{ prime}\}$

Theorem (Hirvensalo, M. & Yakaryılmaz, 2017)

POLY $\notin$ BAfL

PRIME $\notin$ BAfL

Proof idea:

Regularity of matrix multiplication cannot express such languages

NON-AFL LANGUAGES

P a polynomial with nonnegative coefficients and $\deg(P) > 2$,
POLY $= \{a^{P(n)} \mid n \in \mathbb{N}\}$.

PRIME $= \{a^p \mid p \text{ prime}\}$

Theorem (Hirvensalo, M. & Yakaryılmaz, 2019)

POLY $\notin$ $\text{AfL}_{\mathbb{A}}$

PRIME $\notin$ $\text{AfL}_{\mathbb{A}}$

Proof idea:

Same but with finer analysis

Non-linear automaton model inspired by quantum ones

- → Can be simulated by logspace Turing machines
- → But cannot recognize all logspace languages

What next ?

- → Finally prove that POLY, PRIME $\notin$ AfL

We only have POLY, PRIME $\notin \mathrm{AfL}_{\mathbb{A}}$

Non-linear automaton model inspired by quantum ones

- → Can be simulated by logspace Turing machines
- → But cannot recognize all logspace languages

What next ?

- → Finally prove that POLY, PRIME $\notin$ AfL
- → Separation between BAfL and AfL?

For stochastic and quantum, separation based on BSL $=$ REG

Non-linear automaton model inspired by quantum ones

- → Can be simulated by logspace Turing machines
- → But cannot recognize all logspace languages

What next ?

- → Finally prove that POLY, PRIME $\notin$ AfL
- → Separation between BAfL and AfL?
- → Fully characterize BAfL and AfL languages

  Semi-linear languages ?

Big class for which our non-membership proof fails

Thank you !

`HSRM Theme` is released under ›**GNU Public License**‹ : you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.