Affine automata verifiers

Aliya Khadieva and Abuzer Yakaryılmaz

email: aliya.khadi@gmail.com, *arXiv:2104.11192*

LATVIJAS UNIVERSITĀTE ANNO 1919 UNIVERSITY OF LATVIA

Kazan Federal UNIVERSITY

Introduction

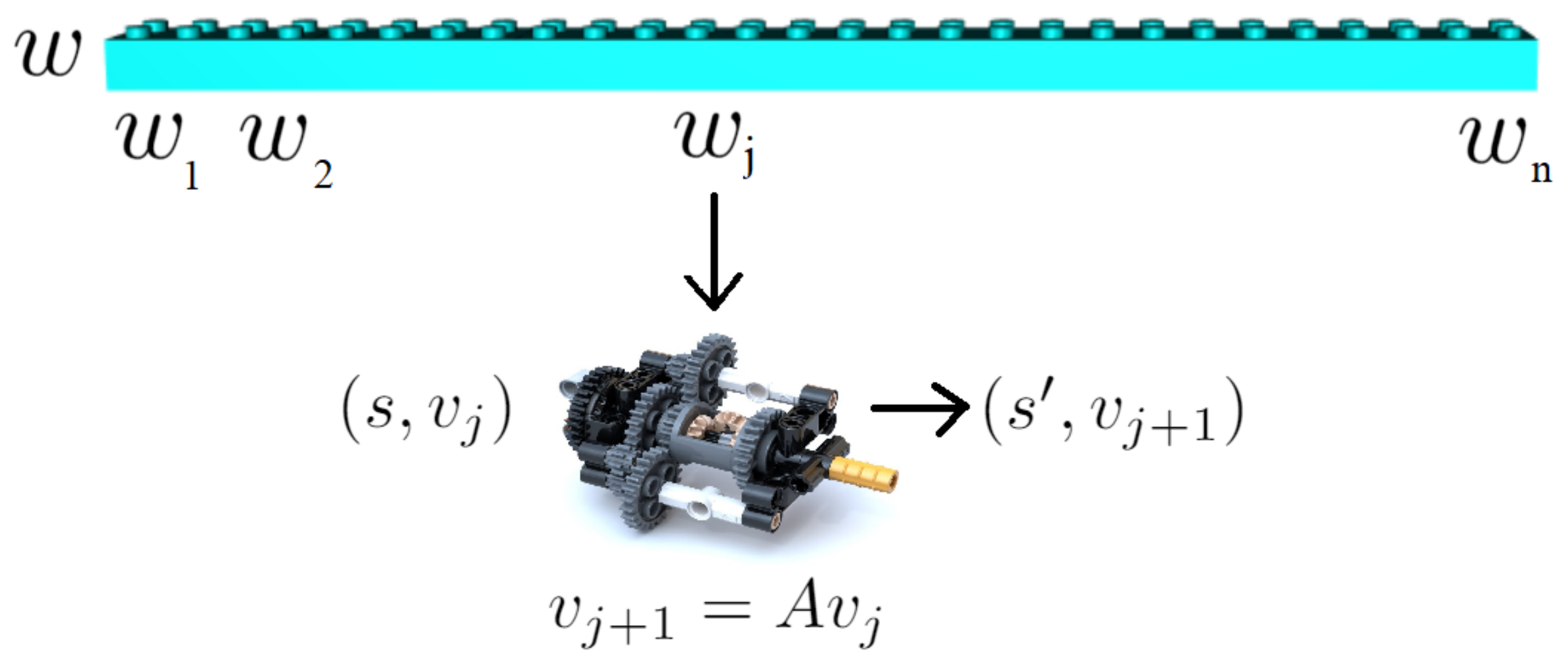
We initiate the study of the verification power of Affine finite automata (AfA) as a part of Arthur-Merlin (AM) proof systems. We show that every unary language is verified by a real-valued AfA verifier. Then, we focus on the verifiers restricted to have only integer-valued or rational-valued transitions. We observe that rational-valued verifiers can be simulated by integer-valued verifiers, and their protocols can be simulated in nondeterministic polynomial time. We show that this bound is tight by presenting an AfA verifier for NP-complete problem SUBSETSUM. We also show that AfAs can verify certain non-affine and non-stochastic unary languages.

An m-state affine system

- An affine state of a system is represented by an m-dimensional vector: $v = (\alpha_1 \quad \cdots \quad \alpha_m)^\intercal \in \mathbb{R}^m$ satisfying that $\sum_{j=1}^{m} \alpha_j = 1$, where α_j, similar to the amplitudes in quantum systems, is the value of the system being in state j
- Any affine operator of this system is a linear operator represented by an $(m \times m)$-dimensional matrix A satisfying that $\sum_{j=1}^{m} a_{j,i} = 1$ for each column i (the column summation is 1). When the operator A is applied to the affine state v, the new state is $v' = A \cdot v$
- To retrieve information from the affine system, similar to the measurement operators of quantum system, we apply a weighting operator. When the affine state v is weighted, the i-th state is observed with probability
$$\frac{|\alpha_i|}{|v|} = \frac{|\alpha_i|}{|\alpha_1| + \cdots + |\alpha_m|}.$$

Finite automata with deterministic and affine states (ADfA)

ADfA is an n-state deterministic finite automaton having an m-state affine register, where $m, n > 0$.
Let $S = \{s_1, \ldots, s_n\}$ be the classical deterministic and $E = \{e_1, \ldots, e_m\}$ be the affine states.
The computation of M is traced by a pair (s, v) called a configuration, where $s \in S$ is the classic state, $v \in \mathbb{R}^m$ is a vector of the affine configuration.



After reading the whole input,

- If the final classical state is not accepting, then reject the input.
- Otherwise, a weighting operator is applied.

  The input is accepted if an affine accepting state is observed. Then, the accepting probability by the affine part is
$$f_M(w) = \frac{\sum_{e_i \in E_a} |v_f[i]|}{|v_f|} \in [0, 1].$$

Acknowledgements

Yakaryılmaz was partially supported by the ERDF project Nr. 1.1.1.5/19/A/005 "Quantum computers with constant memory". A part of research is funded by the subsidy allocated to Kazan Federal University for the state assignment in the sphere of scientific activities, project No. 0671-2020-0065. A part of the reported study is funded by RFBR according to the research project No.20-37-70080.

Affine automata verifiers

In [2], Arthur-Merlin systems with probabilistic finite automata verifier is defined as an automata having both nondeterministic and probabilistic states. We follow the same framework here. We indeed give the ability of making nondeterministic transitions to the model ADfA. A finite automaton with nondeterministic and affine states (ANfA) with n classic nondeterministic and m affine states can have one or more transitions for each configuration.
We define $\mathsf{AM(AfA)}$ as the class of languages verifiable by bounded-error Arthur-Merlin system having realtime affine finite verifiers.

Verification of every unary language

Every unary language $L \subseteq \{a\}^*$ is verified by an ANfA V with error bound 0.155.
Let $L \subseteq \Sigma^*$ be an arbitrary unary language, where $\Sigma = \{a\}$. We define a real number to encode the whole membership information of L as follows: $\alpha_L = \sum_{i=0}^{\infty} \frac{b_i}{32^{i+1}} = \frac{b_0}{32} + \frac{b_1}{32^2} + \frac{b_2}{32^3} + \cdots$, where $b_i = 1$ if $a^i \in L$ and $b_i = 0$ if $a^i \notin L$. In binary form: $bin(\alpha_L) = 0.0000b_0 0000b_1 \cdots 0000b_i \cdots$. Moreover, we define $\alpha_L[j] = \frac{b_j}{32} + \frac{b_{j+1}}{32^2} + \frac{b_{j+2}}{32^3} + \cdots$, where $j \geq 0$.

1. For any $\alpha_L[j]$, there is a unary language $L^{'}$ such that $\alpha_L[j] = \alpha_{L'}$.
2. The values of α_L and so $\alpha_L[j]$ are bounded: $0 \leq \alpha_L \leq \frac{1}{31}$ and $0 \leq \alpha_L[j] \leq \frac{1}{31}$.
3. The values of $\alpha_L[j+1]$ and $\alpha_L[j]$ can be related:
   - If $b_j = 0$: $\alpha_L[j+1] = 32 \cdot \alpha_L[j]$.
   - If $b_j = 1$: $\alpha_L[j+1] = 32 \cdot \alpha_L[j] - 1$.

By using α_L, we design a bounded error ANfA for language L. The main idea behind the protocol is that each b_i is nondeterministically guessed and the verification is done by subtracting the guessed b_i and the actual value b_i encoded in α_L. As long as the nondeterministic choices are correct, the result of such subtractions will be zero. Otherwise, it will not be zero, based on which we reject the input.

Results

On unary languages, for the real-valued verifiers, we show that AfAs and 2QCFAs have the same verification power: $\mathsf{UALL} = \mathsf{UAM(2QCFA)} = \mathsf{UAM(AfA)}$, where AfAs are realtime machines but 2QCFAs run in exponential expected time.
On unary languages, for the rational-valued verifiers, we know that
$$\mathsf{UnaryREG} = \mathsf{UAM}_{\mathbb{Q}}\mathsf{(PFA)} \subseteq \begin{array}{l} \mathsf{UAM}_{\mathbb{Q}}\mathsf{(QFA)} \subseteq \mathsf{UAM(QFA)} \\ \mathsf{UAM}_{\mathbb{Q}}\mathsf{(2PFA)} \subseteq \mathsf{UAM(2PFA)} \end{array},$$
where it is open if the inclusions are strict, and we show that $\mathtt{UPOLY(P)} \in \mathsf{UAM}_{\mathbb{Q}}\mathsf{(AfA)}$ and so we have $\mathsf{UnaryREG} \subsetneq \mathsf{UAM}_{\mathbb{Q}}\mathsf{(AfA)}$.
On non-unary languages, for the rational-valued verifiers, we give an upper bound for $\mathsf{AM}_{\mathbb{Q}}\mathsf{(AfA)}$, and so we have $\mathsf{AM}_{\mathbb{Q}}\mathsf{(AfA)} = \mathsf{AM}_{\mathbb{Z}}\mathsf{(AfA)} \subseteq \mathsf{NP} \cap \mathsf{SPACE(n)} \subsetneq \mathsf{AM}_{\mathbb{Q}}\mathsf{(2QCFA)}$, where 2QCFAs run in double-exponential expected time. Our bound is tight since we show that $\mathtt{SUBSETSUM} \in \mathsf{AM}_{\mathbb{Z}}\mathsf{(AfA)}$.

References

[1] A. Díaz-Caro and A. Yakaryılmaz. 2016. Affine computation and affine automaton. In Computer Science — Theory and Applications (LNCS, Vol. 9691).emY2013 Abuzer Yakaryılmaz. 2013. Public qubits versus private coins. InThe Proceedings of Workshop on Quantum and Classical Complexity. Univeristy ofLatvia Press, 45–60. ECCC:TR12-130.

[2] Anne Condon, Lisa Hellerstein, Samuel Pottle, and Avi Wigderson. 1998. On the Power of Finite Automata with Both Nondeterministic and Probabilistic States.SIAM J. Comput.27, 3 (1998), 739–762