Deciding FO-definability of Regular Languages

Agi Kurucz[1], Vladislav Ryzhikov[2], Yury Savateev[2], Michael Zakharyaschev[2]

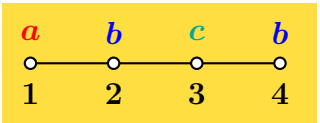
RAMICS 2021

1 King's College London

2 Birkbeck, University of London

FO-definability of Regular Languages

A word $w \in \Sigma^*$ is a first order structure with binary predicate $<$ and unary predicates $a(x)$ for each $a \in \Sigma$ on the domain $[1, |w|]$.



A language can be defined by a formula.

E.g. the formula $\exists x \exists y (x < y) \wedge \mathrm{a}(x) \wedge \mathrm{b}(y)$ defines the language $\Sigma^* \mathrm{a} \Sigma^* \mathrm{b} \Sigma^*$.

FO($<$)-formulas define precisely all star-free languages.

(McNaughton and Papert, '71)

Determining FO($<$)-definability of regular languages is PSPACE-complete.

(Cho, Huynh '91), (Bernátsky '97).

Problem: How Hard is a Given Regular Language?

Let **L** be a regular language.

Does **L** $\in$ AC^0? Does **L** $\in$ ACC^0? Is **L** NC^1-complete?

(Barrington et al. '92) gives the following corresponding descriptive complexities for regular languages:

- AC^0 is equivalent to FO$(<, \equiv)$, which is FO$(<)$ with predicates $x \equiv_p y$ that mean $x - y = 0 \mod p$,
- ACC^0 is equivalent to FO$(<, \mathbf{MOD})$, which is FO$(<)$ with quantifiers $\exists_p^r$, where $\exists_p^r x \phi(x)$ means that the number of positions satisfying $\phi(x)$ is equal to $r \mod p$.

Only decidability of these problems was known previously.

New results: FO$(<, \equiv)$- and FO$(<, \mathbf{MOD})$-definability are PSPACE-complete for regular languages given as DFA, NFA, or 2NFA.

Regular Languages and Monoids

A *monoid* $(M, \cdot, e)$ is a set M with associative operation $\cdot$ and the neutral element e. E.g. Σ^* is a monoid w.r.t. concatenation and the empty word.

A language $\mathbf{L} \subseteq \Sigma^*$ is *recognisable* by a monoid M iff $\mathbf{L} = f^{-1}(f(\mathbf{L}))$ for some morphism $f : \Sigma^* \to M$. Regular languages are exactly the languages recognisable by finite monoids.

In a DFA $\mathcal{A}$ every letter is a function on the set of states. They generate a *transitional monoid* $M(\mathcal{A})$ by composition. It recognizes $\mathbf{L}(\mathcal{A})$.

Definability, Algebraic Characterisations, and Circuit Complexity

Let **L** be a regular language.

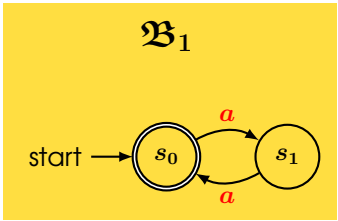
definability of **L**	algebraic characterisation of **L**	circuit complexity
FO($<$)	no non-trivial groups in $M(\mathbf{L})$	in AC^0
FO($<, \equiv$)	no non-trivial groups in $\eta_L(\Sigma^t)$ for any t	
FO($<, \mathbf{MOD}$)	no unsolvable groups in $M(\mathbf{L})$	in ACC^0
not in FO($<, \mathbf{MOD}$)	$M(\mathbf{L})$ contains an unsolvable group	NC^1-complete

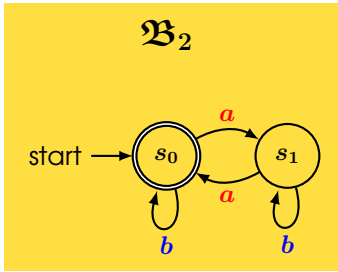
$M(\mathbf{L})$ is the transitional monoid of the minimal DFA recognising **L**.

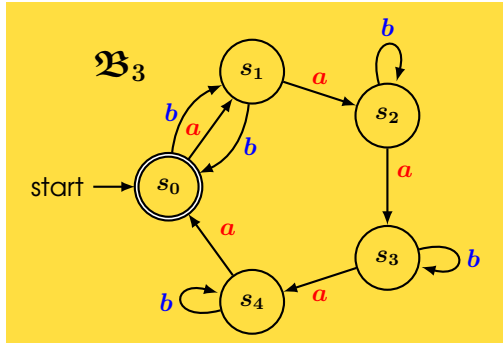
$\eta_{\mathbf{L}} : \Sigma^* \to M(\mathbf{L})$ is the syntactic morphism.

(Barrington et al. '92).

Examples







- $M(\mathfrak{B}_1) \sim \mathbb{Z}_2$, therefore $\mathbf{L}(\mathfrak{B}_1) = \{a^{2n} \mid n \in \mathbb{N}\}$ is not FO(<)-definable (star-free). It is in AC^0.

- $M(\mathfrak{B}_2) \sim \mathbb{Z}_2$, but also $M(\mathfrak{B}_2) = \eta_{\mathbf{L}(\mathfrak{B}_2)}(\{a, b\})$, so $\mathbf{L}(\mathfrak{B}_2) = \{v \mid |v|_a \text{ is even}\}$ is not FO(<, ≡)-definable, not in AC^0, but is in ACC^0.

- $M(\mathfrak{B}_3) \sim S_5$, so $\mathbf{L}(\mathfrak{B}_3)$ is not in FO(<,**MOD**) and is NC^1-complete.

Non-Trivial Groups and How to Find Them (in a Monoid)

Any non-trivial group: find x such that $x = x^{n+1} \neq x^n$ for some $n > 1$.

A non-trivial group in $\eta_{\mathsf{L}}(\Sigma^t)$: find $x, e \in \eta_{\mathsf{L}}(\Sigma^t)$ such that $x = x^{n+1} \neq x^n = e$ for some $n > 1$.

An unsolvable group: find distinct $x, y, e \in M(\mathbf{L})$ and coprime odd $k, l > 1$ such that $e = x^2 = y^k = (xy)^l$, $xe = x$, and $ye = y$.
(See (Kaplan, Levy 2010)).

PSPACE

For a given automaton $\mathcal{A}$ the monoid $M(\mathbf{L}(\mathcal{A}))$ can have exponentially many elements. (Holzer and König '04)

Every monoid recognising $\mathbf{L}$ has $M(\mathbf{L})$ as a factor monoid: $M(\mathbf{L}) = M(\mathcal{A})/\sim_{\mathbf{L}}$.

We can store elements of $M(\mathcal{A})$, multiply, and check equivalence w.r.t. $\sim_{\mathbf{L}}$ in polynomial space (even for 2NFAs).

Therefore all the problems considered here are in PSPACE.

PSPACE-hardness

It is sufficient to show PSPACE-hardness only for minimal DFAs.

Let $\mathfrak{M}$ be a deterministic Turing machine that decides a language using at most $N = P_{\mathfrak{M}}(n)$ tape cells on any input of size n, for some polynomial $P_{\mathfrak{M}}$.

Given $\mathfrak{M}$ and an input w, we define three minimal DFAs whose languages are, respectively, FO($<$)-, FO($<, \equiv$)-, and FO($<, \mathbf{MOD}$)-definable iff $\mathfrak{M}$ rejects w.

The Construction

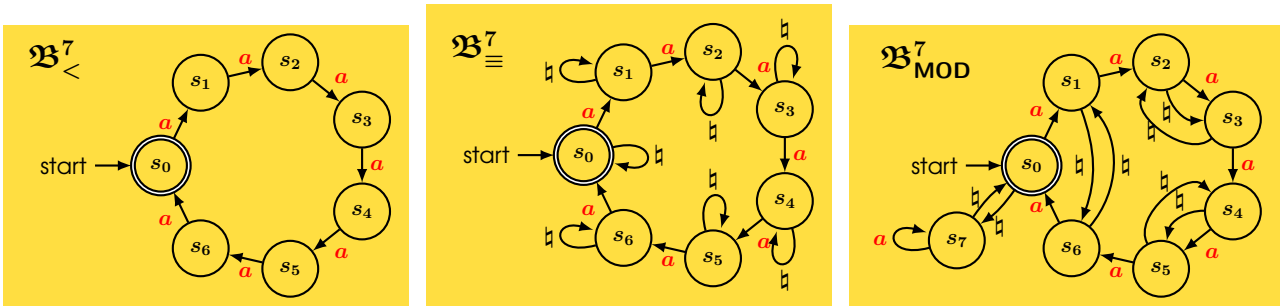
We define a series of minimal DFAs $\mathcal{A}_i$ for $i \in [0, p]$ such that the following holds:

- p is a prime number bigger than $N + 1$
- $p \not\equiv \pm 1 \pmod{10}$.
- All of the languages $\mathsf{L}(\mathcal{A}_i)$, for $i \leq p$, are FO($<$)-definable (star-free).
- $\mathfrak{M}$ accepts w iff $\bigcap_{i=0}^{p} \mathsf{L}(\mathcal{A}_i) \neq \emptyset$.

We can use the construction from (Cho, Huynh '91) here, but in the paper we use a more streamlined one.

The Construction cont.

Next, we require three DFAs $\mathfrak{B}^p_<$, $\mathfrak{B}^p_\equiv$ and $\mathfrak{B}^p_{\mathbf{MOD}}$; see the picture below for $p = 7$.



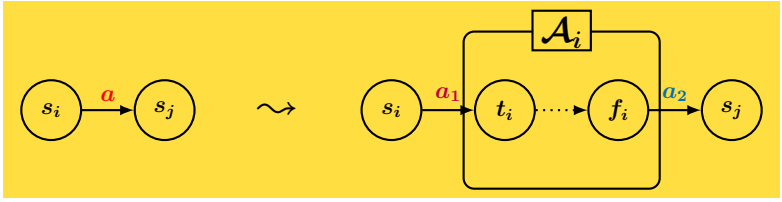
Formally in $\mathfrak{B}^p_{\mathbf{MOD}}$ for $i \in [1, p-1]$ $\delta_↯(s_i) = s_j$ where $j = -1/i$ in the field F_p.

$M(\mathfrak{B}^p_{\mathbf{MOD}})$ is isomorphic to $\mathbf{PSL_2}(p)$ which is unsolvable, but all its proper subgroups are (King '77).

Finishing the Construction

Finally, we define the three automata $\mathcal{A}_<$, $\mathcal{A}_\equiv$, $\mathcal{A}_{\mathsf{MOD}}$.

We take, respectively, $\mathfrak{B}^p_<$, $\mathfrak{B}^p_\equiv$, $\mathfrak{B}^p_{\mathsf{MOD}}$ and replace each transition $s_i \to_a s_j$ by a fresh copy of $\mathcal{A}_i$:



It follows that $\mathcal{A}_<$, $\mathcal{A}_\equiv$, and $\mathcal{A}_{\mathsf{MOD}}$ are minimal DFAs of size polynomial in N, $|\mathfrak{M}|$.

The Main Theorem

- $\mathbf{L}(\mathcal{A}_<)$ is FO($<$)-definable iff $\bigcap_{i=0}^{p} \mathbf{L}(\mathcal{A}_i) = \emptyset$.
- $\mathbf{L}(\mathcal{A}_\equiv)$ is FO($<, \equiv$)-definable iff $\bigcap_{i=0}^{p} \mathbf{L}(\mathcal{A}_i) = \emptyset$.
- $\mathbf{L}(\mathcal{A}_{\mathbf{MOD}})$ is FO($<, \mathbf{MOD}$)-definable iff $\bigcap_{i=0}^{p} \mathbf{L}(\mathcal{A}_i) = \emptyset$.

So FO($<, \equiv$)- and FO($<, \mathbf{MOD}$)-definability of regular languages are PSPACE-hard.

Thank you for your attention!