Deciding FO-definability of Regular Languages

Agi Kurucz\(^1\), Vladislav Ryzhikov\(^2\), Yury Savateev\(^2,3\), and Michael Zakharyaschev\(^2,3\)

\(^1\) King’s College London, UK
\(^2\) Birkbeck, University of London, UK
\(^3\) HSE University, Moscow, Russia

Abstract. We prove that, similarly to known PSPACE-completeness of recognising FO(\(<\))-definability of the language \(L(\mathfrak{A})\) of a DFA \(\mathfrak{A}\), deciding both FO(\(<,\equiv\))- and FO(\(<,\text{MOD}\))-definability (corresponding to circuit complexity in AC\(^0\) and ACC\(^0\)) are PSPACE-complete. We obtain these results by first showing that known algebraic characterisations of FO-definability of \(L(\mathfrak{A})\) can be captured by ‘localisable’ properties of the transition monoid of \(\mathfrak{A}\). Using our criterion, we then generalise the known proof of PSPACE-hardness of FO(\(<\))-definability, and establish the upper bounds not only for arbitrary DFAs but also for 2NFAs.

1 Introduction

This paper gives answers to some open questions related to finite automata, logic and circuit complexity. Research in this area goes back (at least) to the early 1960s when Büchi [8], Elgot [12] and Trakhtenbrot [28] showed that MSO(\(<\)) (monadic second-order) sentences over finite strict linear orders define exactly the class of regular languages.

FO(\(<\))-definable regular languages were proven to be the same as star-free languages [19], and their algebraic characterisation as languages with aperiodic syntactic monoids was obtained in [23]. Algebraic characterisations of FO-definability in other signatures, and circuit and descriptive complexity of regular languages were investigated in [3, 4, 26], which established an AC\(^0\)/ACC\(^0\)/NC\(^1\) trichotomy. In particular, the regular languages decidable in AC\(^0\) are definable by FO(\(<,\equiv\))-sentences with unary predicates \(x \equiv 0 \mod n\); those in ACC\(^0\) are definable by FO(\(<,\text{MOD}\))-sentences with quantifiers \(\exists^* x \psi(x)\) checking whether the number of positions satisfying \(\psi\) is divisible by \(n\); and all regular languages are definable in FO(RPR) with relational primitive recursion [11]; see Table 1.

The problem of deciding whether the language of a given DFA \(\mathfrak{A}\) is FO(\(<\))-definable is known to be PSPACE-complete [7,10,25] (which is also a special case of general results on finite monoids [5,13]). As shown in [4], the algebraic criteria of Table 1 yield algorithms deciding whether a given regular language is in AC\(^0\) and FO(\(<,\equiv\))-definable, or in ACC\(^0\) and FO(\(<,\text{MOD}\))-definable, or NC\(^1\)-complete and is not FO(\(<,\text{MOD}\))-definable (unless ACC\(^0\) = NC\(^1\)). However, these ‘brute force’ algorithms are not optimal, requiring the generation of the
whole transition monoid of $\mathfrak{A}$, which can be of exponential size [14]. As far as we know, the precise complexity of these decision problems has remained open.

Our interest in the exact complexity of these problems is motivated by recent advances in ontology-based data access (OBDA) with linear time temporal logic $LTL$ [11,2]. The classical (atemporal) OBDA paradigm [20,30] relies on a reduction of answering a query mediated by an ontology under the open-world semantics to evaluating a database query in a standard language such as SQL or its extension—that is, essentially, an extension of first-order logic—under the closed-world semantic. In the context of temporal OBDA, answering $LTL$ ontology-mediated queries is equivalent to deciding certain regular languages given by an NFA or 2NFA of (possibly) exponential size, which gives rise to the circuit complexity and FO-definability problems for those languages. For further details the reader is referred to [22], which relies on the results we obtain below.

Our contribution in this paper is as follows. Let $L$ be one of the languages $\text{FO}(\langle, \equiv \rangle)$ or $\text{FO}(\langle, \text{MOD} \rangle)$. First, using the algebraic characterisation results of [3,4,26], we obtain criteria for the $L$-definability of the language $L(\mathfrak{A})$ of any given DFA $\mathfrak{A}$ in terms of a limited part of the transition monoid of $\mathfrak{A}$ (Theorem 1). Then, by using our criteria and generalising the construction of [10], we show that deciding $L$-definability of $L(\mathfrak{A})$ for any minimal DFA $\mathfrak{A}$ is PSpace-hard (Theorem 2). Finally, we apply our criteria to give a PSpace-algorithm deciding $L$-definability of $L(\mathfrak{A})$ for not only any DFA but any 2NFA $\mathfrak{A}$ (Theorem 3).

2 Preliminaries

We begin by briefly reminding the reader of the basic algebraic and automata-theoretic notions required in the remainder of the paper.

2.1 Monoids and Groups

A semigroup is a structure $\mathcal{S} = (S, \cdot)$ where $\cdot$ is an associative binary operation. Given $s, s' \in S$ and $n > 0$, we write $s^n$ for $s \cdot \ldots \cdot s$ $n$-times, and often write $ss'$ for $s \cdot s'$. An element $s$ in a semigroup $\mathcal{S}$ is idempotent if $s^2 = s$. An element $e$ in $\mathcal{S}$ is an identity if $e \cdot x = x \cdot e = x$ for all $x \in S$. (It is easy to see that such

| definability of $L$ | algebraic characterisation of $L$ | circuit complexity |
|---------------------|----------------------------------|--------------------|
| $\text{FO}(\langle \rangle)$ | $M(L)$ is aperiodic | in $\text{AC}^0$ |
| $\text{FO}(\langle, \equiv \rangle)$ | $\eta_L$ is quasi-aperiodic | |
| $\text{FO}(\langle, \text{MOD} \rangle)$ | all groups in $M(L)$ are solvable | in $\text{ACC}^0$ |
| $\text{FO}(\text{RPR})$ | arbitrary $M(L)$ | in $\text{NC}^1$ |

| not in $\text{FO}(\langle, \text{MOD} \rangle)$ | $M(L)$ contains an unsolvable group | $\text{NC}^1$-hard |

Table 1. Definability, algebraic characterisations and circuit complexity of a regular language $L$, where $M(L)$ is the syntactic monoid and $\eta_L$ the syntactic morphism of $L$. 
Here, we only need the fact that any homomorphic image of a solvable group is solvable and the Kaplan–Levy criterion \[16\] (generalising Thompson’s \[27, \]

For a definition of solvable and unsolvable groups the reader is referred to \[21\]. Here, we only need the fact that any homomorphic image of a solvable group is solvable and the Kaplan–Levy criterion \[16\] (generalising Thompson’s \[27\].
Cor. 3) according to which a finite group $G$ is unsolvable if it contains three elements $a, b, c$, such that $o_G(a) = 2$, $o_G(b)$ is an odd prime, $o_G(c) > 1$ and coprime to both 2 and $o_G(b)$, and $abc$ is the identity element of $G$.

A one-to-one and onto function on a finite set $S$ is called a permutation on $S$. The order of a permutation $\delta$ is its order in the group of all permutations on $S$ (whose operation is composition, and its identity element is the identity permutation $\text{id}_S$). We use the standard cycle notation for permutations.

Suppose $G$ is a monoid of $Q \to Q$ functions, for some finite set $Q \neq \emptyset$. Let $s = \{ q \in Q \mid e_G(q) = q \}$, where $e_G$ the identity element in $G$. For every function $\delta$ in $G$, let $\delta|_s$ denote the restriction of $\delta$ to $S$. Then we have the following:

$G$ is a group iff $\delta|_s$ is a permutation on $S$, for every $\delta$ in $G$; \hspace{1cm} (8)

if $G$ is a group and $\delta$ is a nonidentity element in it, then $\delta|_s \neq \text{id}_S$ and the order of the permutation $\delta|_s$ divides $o_G(\delta)$. \hspace{1cm} (9)

### 2.2 Automata: DFAs, NFAs, 2NFAs

A two-way nondeterministic finite automaton is a quintuple $\mathfrak{A} = (Q, \Sigma, \delta, Q_0, F)$ that consists of an alphabet $\Sigma$, a finite set $Q$ of states with a subset $Q_0 \neq \emptyset$ of initial states and a subset $F$ of accepting states, and a transition function $\delta: Q \times \Sigma \to 2^Q \times \{-1, 1\}$ indicating the next state and whether the head should move left ($-1$), right ($1$), or stay put. If $Q_0 = \{ q_0 \}$ and $|\delta(q, a)| = 1$, for all $q \in Q$ and $a \in \Sigma$, then $\mathfrak{A}$ is deterministic, in which case we write $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$. If $\delta(q, a) \subseteq Q \times \{ 1 \}$, for all $q \in Q$ and $a \in \Sigma$, then $\mathfrak{A}$ is a one-way automaton, and we write $\delta: Q \times \Sigma \to 2^Q$. As usual, DFA and NFA refer to one-way deterministic and non-deterministic finite automata, respectively, while 2DFA and 2NFA to the corresponding two-way automata. Given a 2NFA $\mathfrak{A}$, we write $q \to_{a, \delta} q'$ if $(q', d) \in \delta(q, a)$; given an NFA $\mathfrak{A}$, we write $q \to_a q'$ if $q' \in \delta(q, a)$. A run of a 2NFA $\mathfrak{A}$ is a word in $(Q \times \mathbb{N})^*$. A run $(q_0, i_0), \ldots, (q_m, i_m)$ is a run of $\mathfrak{A}$ on a word $w = a_0 \ldots a_n \in \Sigma^*$ if $q_0 \in Q_0$, $i_0 = 0$ and there exist $d_0, \ldots, d_{n-1} \in \{-1, 0, 1\}$ such that $q_j \to_{a_i, d_i} q_{j+1}$ and $i_{j+1} = i_j + d_j$ for all $j$, $0 \leq j < m$. The run is accepting if $q_m \in F$, $i_m = n + 1$. $\mathfrak{A}$ accepts $w \in \Sigma^*$ if there is an accepting run of $\mathfrak{A}$ on $w$; the language $L(\mathfrak{A})$ of $\mathfrak{A}$ is the set of all words accepted by $\mathfrak{A}$.

Given an NFA $\mathfrak{A}$, states $q, q' \in Q$, and $w = a_0 \ldots a_n \in \Sigma^*$, we write $q \to_w q'$ if either $w = \varepsilon$ and $q' = q$ or there is a run of $\mathfrak{A}$ on $w$ that starts with $(q_0, 0)$ and ends with $(q', n + 1)$. We say that a state $q \in Q$ is reachable if $q' \to_w q$, for some $q' \in Q_0$ and $w \in \Sigma^*$.

Given a DFA $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$ and a word $w \in \Sigma^*$, we define a function $\delta_w: Q \to Q$ by taking $\delta_w(q) = q'$ if $q \to_w q'$. We also define an equivalence relation $\sim$ on the set $Q^* \subseteq Q$ of reachable states by taking $q \sim q'$ if, for every $w \in \Sigma^*$, we have $\delta_w(q) = F$ just in case $\delta_w(q') = F$. We denote the $\sim$-class of $q$ by $q/\sim$, and let $X/\sim = \{ q/\sim \mid q \in X \}$ for any $X \subseteq Q^*$. Define $\delta_w: Q^*/\sim \to Q^*/\sim$ by taking $\delta_w(q/\sim) = \delta_w(q)/\sim$. Then $(Q^*/\sim, \Sigma, \delta, q_0/\sim, (F \cap Q^*)/\sim)$ is the minimal DFA whose language coincides with the language of $\mathfrak{A}$. Given a regular language $L$, we denote by $\mathfrak{A}_L$ the minimal DFA whose language is $L$. 
The transition monoid of a DFA $A$ is $M(A) = \{\delta_w \mid w \in \Sigma^\ast\}$, with $\delta_v \cdot \delta_w = \delta_{vw}$ for any $v, w$. The syntactic monoid $M(L)$ of $L$ is the transition monoid $M(A_L)$ of $A_L$. The syntactic morphism of $L$ is the map $\eta_L$ from $\Sigma^\ast$ to the domain of $M(L)$ defined by $\eta_L(w) = \delta_w$. We call $\eta_L$ quasi-aperiodic if $\eta_L(\Sigma^\ast)$ is aperiodic for every $L < \omega$.

Suppose $L \in \{\text{FO}(\prec), \text{FO}(\prec, \equiv), \text{FO}(\prec, \text{MOD})\}$. A language $L$ over an alphabet $\Sigma$ is $L$-definable if there is an $L$-sentence $\varphi$ in the signature $\Sigma$, whose symbols are treated as unary predicates, such that, for any $w \in \Sigma^\ast$, we have $w = a_0 \ldots a_n \in L$ iff $\forall w \models \varphi$, where $\forall w$ is an FO-structure with domain $\{0, \ldots, n\}$ ordered by $\prec$, in which $\forall w \models a(i)$ iff $a = a_i$, for $0 \leq i \leq n$.

Table 1 summarises the known results that connect definability of a regular language $L$ with properties of the syntactic monoid $M(L)$ and syntactic morphism $\eta_L$ (see [1] for details) and with its circuit complexity under a reasonable binary encoding of $L$’s alphabet (see, e.g., [7] Lemma 2.1) and the assumption that $\text{ACC}^0 \neq \text{NC}^1$. We also remind the reader that a regular language is $\text{FO}(\prec)$-definable iff it is star-free [26], and that $\text{AC}^0 \nsubseteq \text{ACC}^0 \subseteq \text{NC}^1$ [15,29].

3 Criteria of $L$-definability

In this section, we show that the algebraic characterisations of $\text{FO}$-definability of $L(A)$ given in Table 1 can be captured by ‘localisable’ properties of the transition monoid of $A$, for any given DFA $A$. Note that Theorem 1(i) was already observed in [25] and used in proving that $\text{FO}(\prec)$-definability of $L(A)$ is $\text{PSPACE}$-complete [7,10,26]; while criteria (ii) and (iii) seem to be new.

**Theorem 1.** For any DFA $A = (Q, \Sigma, \delta, q_0, F)$, the following criteria hold:

(i) $L(A)$ is not $\text{FO}(\prec)$-definable iff $A$ contains a nontrivial cycle, that is, there exist a word $u \in \Sigma^\ast$, a state $q \in Q^\ast$, and a number $k \leq |Q|$ such that $q \neq \delta_u(q)$ and $q = \delta_v(q)$;

(ii) $L(A)$ is not $\text{FO}(\prec, \equiv)$-definable iff there are words $u, v \in \Sigma^\ast$, a state $q \in Q^\ast$, and a number $k \leq |Q|$ such that $q \neq \delta_u(q)$, $q = \delta_v(q)$, $|v| = |u|$, and $\delta_{u_i}(q) = \delta_{v_i}(q)$, for every $i < k$;

(iii) $L(A)$ is not $\text{FO}(\prec, \text{MOD})$-definable iff there exist words $u, v \in \Sigma^\ast$, a state $q \in Q^\ast$ and numbers $k, l \leq |Q|$ such that $k$ is an odd prime, $l > 1$ and coprime to both $2$ and $k$, $q \neq \delta_u(q)$, $q \neq \delta_v(q)$ and, for all $x \in \{u, v\}^*$, we have $\delta_x(q) = \delta_{ux^k}(q) = \delta_{ux^k}(q) = \delta_{ux^k}^{-1}(q)$.

**Proof.** Throughout, we use the algebraic criteria of Table 1 for $L = L(A)$. Thus, $M(L)$ is the transition monoid of the minimal DFA $\mathfrak{N}(M(L))$, whose transition function we denote by $\delta$.

(i) ($\Rightarrow$) Suppose $\Phi$ is a nontrivial group in $M(\mathfrak{N}(M(L)))$. Let $u \in \Sigma^\ast$ be such that $\delta_u$ is a nonidentity element in $\Phi$. We claim that there is $p \in Q^\ast$ such that $\delta_{u^p}(p/\sim) \neq \delta_{u^{p+1}}(p/\sim)$ for any $n > 0$. Indeed, otherwise for every $p \in Q^\ast$ there is $n_p > 0$ with $\delta_{u^{n_p}}(p/\sim) = \delta_{u^{n_p+1}}(p/\sim)$. Let $n = \max\{n_p \mid p \in Q^\ast\}$. Then $\delta_u = \delta_{u^{n+1}}$, contrary to (1).
By (5), there is $m \geq 1$ with $\bar{\delta}_{\bar{u}^m}(p_{/\bar{u}}) = \bar{\delta}_{u^m}(p_{/\bar{u}})$. Let $s_{/\sim} = \bar{\delta}_{u^m}(p_{/\bar{u}})$. Then $s_{/\sim} = \bar{\delta}_{u^m}(s_{/\sim})$, and so the restriction of $\delta_{u^m}$ to the subset $s_{/\sim}$ of $Q^r$ is an $s_{/\sim}$-function. By (6), there exist $n \in s_{/\sim}$ and $n \geq 1$ such that $(\delta_{u^m})^n(q) = q$. Thus, $\delta_{u^m}(q) = q$, and so by (7), there is $k \leq |Q|$ with $\delta_{u^k}(q) = q$. As $s_{/\sim} \neq \bar{\delta}(s_{/\sim})$, we also have $q \neq \bar{\delta}(q)$, as required.

(i) ($\iff$) Suppose the condition holds for $\mathfrak{X}$. Then there are $u \in \Sigma^*$, $q \in Q'/\sim$, and $k < \omega$ such that $q \neq \delta_u(q)$ and $q = \delta_{u^k}(q)$. Then $\delta_{u^k} \neq \delta_{u^{k+1}}$ for any $n > 0$. Indeed, otherwise we would have some $n > 0$ with $\delta_{u^n}(q) = \delta_{u^{n+1}}(q)$. Let $i, j$ be such that $n = i \cdot k + j$ and $j < k$. Then

$$q = \bar{\delta}_{u^k}(q) = \bar{\delta}_{u^{k+i}}(q) = \bar{\delta}_{u^{n+k}}(q) = \bar{\delta}_{u^{(i+1)n}}(q) = \bar{\delta}_{u}(q).$$

So, by (3) and (4), $\mathfrak{G}_{\delta_u}$ is a nontrivial group in $M(\mathfrak{X}_{\mathfrak{L}(3)})$.

(ii) ($\Rightarrow$) Let $\mathfrak{G}$ be a nontrivial group in $\eta_{\mathfrak{L}}(\Sigma^*)$, for some $t < \omega$, and let $u \in \Sigma^*$ be such that $\delta_u$ is a nonidentity element in $\mathfrak{G}$. As shown in the proof of (i) ($\Rightarrow$), there exist $s \in Q^r$ and $m \geq 1$ such that $s_{/\sim} \neq \bar{\delta}_{u}(s_{/\sim})$ and $s_{/\sim} = \bar{\delta}_{u^m}(s_{/\sim})$. Now let $v \in \Sigma^*$ be such that $\bar{\delta}_v$ is the identity element in $\mathfrak{G}$, and consider $\delta_v$.

By (2), there is $\ell \geq 1$ such that $\delta_{v^i}$ is idempotent. Then $\delta_{v^{i-1}} = \delta_{v^{i-1}}$. Thus, if we let $u = u_0 \ell^{i-1}$ and $v = v^{i-1}$, then $|u| = |v|$ and $\delta_{u^i} = \delta_{u^i}$ for any $i < \omega$. Also, $\delta_{u^i} = \delta_{u^i}$ for every $i \geq 1$, and so the restriction of $\delta_{u^m}$ to $s_{/\sim}$ is an $s_{/\sim}$-function. By (3), there exist $q \in s_{/\sim}$ and $n \geq 1$ such that $(\delta_{u^m})^n(q) = q$. Thus, $\delta_{u^{n+k}}(q) = q$, and so by (6), $\delta_{u^{n+k}}(q) = q$. As $s_{/\sim} \neq \bar{\delta}_{u^m}(s_{/\sim}) = \bar{\delta}_{u}(s_{/\sim})$, we also have $q \neq \bar{\delta}(q)$, as required.

(iii) ($\Leftarrow$) If the condition holds for $\mathfrak{X}$, then there exist $u, v \in \Sigma^*$, $q \in Q'/\sim$, and $k < \omega$ such that $q \neq \bar{\delta}_{u^k}(q)$, $q = \bar{\delta}_{u^k}(q)$, $|v| = |u|$, and $\bar{\delta}_{v^k}(q) = \delta_{u^k}(q)$, for every $i < k$. As $M(\mathfrak{X}_{\mathfrak{L}(3)})$ is finite, it has finitely many subsets. So there exist $i, j \geq 1$ such that $\eta_{\mathfrak{L}}(\Sigma^{|u|}) = \eta_{\mathfrak{L}}(\Sigma^{|v^j|})$. Let $z$ be a multiple of $j$ with $i \leq z < i + j$. Then $\eta_{\mathfrak{L}}(\Sigma^{|u^{|u|}}) = \eta_{\mathfrak{L}}(\Sigma^{|v^{|v|}})$, and so $\eta_{\mathfrak{L}}(\Sigma^{|u|})$ is closed under the composition of functions (that is, the semigroup operation of $M(\mathfrak{X}_{\mathfrak{L}(3)})$). Let $w = w_{v_{i-j}^{-1}}$ and consider the group $\mathfrak{G}_{\delta_u}$ (defined above (2)–(4)). Then $G_{\delta_v} \subseteq \eta_{\mathfrak{L}}(\Sigma^{|u|})$. We claim that $\mathfrak{G}_{\delta_u}$ is nontrivial. Indeed, we have $\bar{\delta}_{u^{|u|}}(q) = \delta_{u^{|u|}}(q) = q$. On the other hand, $\bar{\delta}_{u^{|u|}}(q) = \delta_{u^{|u|}}(q) = q$. By the proof of (i) ($\Leftarrow$), $\mathfrak{G}_{\delta_u}$ is nontrivial.

(iii) (⇐) Suppose $\mathfrak{G}$ is an unsolvable group in $M(\mathfrak{X}_{\mathfrak{L}(3)})$. By the Kaplan–Levy criterion, $\mathfrak{G}$ contains three functions $a, b, c$ such that $\delta_a(2) = 2$, $\delta_b(2)$ is an odd prime, $\delta_b(2) > 1$ and coprime to both $2$ and $\delta_b(2)$, and $c$ is a divisor for the identity element $\delta_e$ of $\mathfrak{G}$. Let $u, v \in \Sigma^*$ be such that $a = \delta_{u^m}, b = \delta_{v^m}$, and $c = (\delta_{u^m})^{-1}$, and let $k = (\delta_{u^m})^{-1}$ and $r = \delta_{u^m}(c) = \delta_{v^m}(c)$. Then $r > 1$ and coprime to both $2$ and $k$. Let $S = \{ p \in Q'/\sim | e_{\mathfrak{G}}(p) = p \}$. As $\delta_x$ is $\mathfrak{G}$ for every $x \in \{ u, v \}$, we have $e_{\mathfrak{G}} \circ \delta_x = \delta_x$. Thus,

$$\bar{\delta}_{v^{|v|}}(q) = \delta_{v^{|v|}}(\delta_{v^{|v|}}(q)) = e_{\mathfrak{G}}(\delta_{v^{|v|}}(q)) = e_{\mathfrak{G}}(\delta_{v^{|v|}}(q)) = \delta_{v^{|v|}}(q),$$

and

$$\bar{\delta}_{u^{|u|}}(q) = \bar{\delta}_{u^{|u|}}(\bar{\delta}_{u^{|u|}}(q)) = e_{\mathfrak{G}}(\bar{\delta}_{u^{|u|}}(q)) = e_{\mathfrak{G}}(\bar{\delta}_{u^{|u|}}(q)) = \delta_{u^{|u|}}(q),$$

for every $q \in S$.

Then, by (3), each of $\delta_{u^{|u|}}, \delta_{v^{|v|}}$ and $\delta_{u^{|u|}}$ is a permutation on $S$. By (4), the order of $\delta_{u^{|u|}}$ is $2$, the order of $\delta_{v^{|v|}}$ is $k$, and the order $l$ of $\delta_{u^{|u|}}$ is a $> 1$ divisor.
of r, and so it is coprime to both 2 and k. Also, we have k, l ≤ |S| ≤ |Q|. Further, for every x, if q is in S then δ_x(q) ∈ S as well. So we have
\[ δ_{x(uv)}(q) = δ_{x(u)}(δ_{x(v)}(q)) = (δ_{uv}|S)(δ_x(q)) = id_S(δ_x(q)) = δ_x(q), \]
for all q ∈ S.

It remains to show that there is q ∈ S with q ≠ δ_u(q), q ≠ δ_u(q), and q ≠ δ_u(q). Recall that the length of any cycle in a permutation divides its order. First, we show there is q ∈ S with q ≠ δ_u(q) and q ≠ δ_u(q). Indeed, as δ_u|S ≠ id_S, there is q ∈ S such that δ_u(q) = q ≠ q. As the order of δ_u|S is 2, δ_u(q') = q. If both δ_u(q) = q and δ_u(q') = q' were the case, then δ_u(q) = q' and δ_u(q') = q would hold, and so (qq') would be a cycle in δ_u|S, contrary to l being coprime to 2. So take some q ∈ S with δ_u(q) = q ≠ q and δ_u(q) ≠ q. If δ_u(q') = q then δ_u(q) ≠ q, and so q is a good choice. Suppose δ_u(q) = q, and let q' ≠ δ_u(q). Then q' ≠ q', as k is odd. Thus, δ_u(q) ≠ q', and so q' is a good choice.

(iii) (≤) Suppose u, v ∈ Σ∗, q ∈ Q∗, and k, l < ω are satisfying the conditions. For every x ∈ {u, v}∗, we define an equivalence relation ≈_x on Q∗/∼ by taking p ≈ x p' if δ_x(p) = δ_x(p'). Then we clearly have that ≈_x ≃ ≈_x, for all x, y ∈ {u, v}∗. As Q is finite, there is z ∈ {u, v}∗ such that ≈_x ≃ ≈_y for all y ∈ {u, v}∗. Take such a z. By (2), δ^2 is idempotent for some n ≥ 1. We let w = z^n. Then δ_u is idempotent and we also have that
\[ ≈_w = ≈_{uw} \]
for all y ∈ {u, v}∗. (10)

Let G_{u,v} = {δ_{uw} \mid x ∈ {u, v}∗}. Then G_{u,v} is closed under composition. Let Θ_{u,v} be the subsemigroup of M(ΘL(3)) with universe G_{u,v}. Then δ_u = δ_{uw} is an identity element in Θ_{u,v}. Let S = {p ∈ Q∗/∼ \mid δ_u(p) = p}. We show that
\[ \text{for every } δ \in Θ_{u,v}, δ|S \text{ is a permutation on } S, \]
and so Θ_{u,v} is a group by (3). Indeed, take some x ∈ {u, v}∗. As δ_u(δ_{uw}(p)) = δ_{uw}(p) for any p ∈ Q∗/∼, δ_{uw}|S is an S → S function. Also, if p, p' ∈ S and δ_{uw}(p) = δ_{uw}(p'), then p ≈_{uw} p'. Thus, by (10), p ≈_{uw} p', that is, p = δ_u(p) = δ_u(p') = p', proving (11).

We show that Θ_{u,v} is unsolvable by finding an unsolvable homomorphic image of it. Let R = {p ∈ Q∗/∼ \mid p = δ_x(q) for some x ∈ {u, v}∗}. We claim that, for every δ in Θ_{u,v}, δ|R is a permutation on R, and so the function h mapping every δ to δ|R is a group homomorphism from Θ_{u,v} to the group of all permutations on R. Indeed, by (11), it is enough to show that R ⊆ R. Let \( \overline{w} = \overline{z}_m \ldots \overline{z}_1 \), where \( w = z_1 \ldots z_m \) for some \( z_i \in \{u, v\} \), \( \overline{w} = u \) and \( u = v^{k-1} \). Since δ_u(q) = δ_{ux}(q) = δ_{ux}(q) for all x ∈ {u, v}∗, we obtain that
\[ δ_{uy}(q) = δ_{ux}(q) = δ_{u}(q), \]
for all q ∈ {u, v}∗. (12)

Now suppose p ∈ R, that is, p = δ_x(q) for some x ∈ {u, v}∗. Then, by (12),
\[ δ_u(p) = δ_u(δ_x(q)) = δ_{ux}(q) = δ_{xuw}(q) = δ_{xuw}(q) = δ_x(q) = p, \]
and so \( p \in S \), as required.

Now let \( \mathcal{G} \) be the image of \( \mathcal{G}_{a,b,v} \) under \( h \). We prove that \( \mathcal{G} \) is unsolvable by finding three elements \( a, b, c \) in it such that \( \sigma(a) = 2 \), \( \sigma(b) = k \), \( \sigma(c) \) is coprime to both \( 2 \) and \( \sigma(b) \), and \( c \circ b \circ a = \text{id}_R \) (the identity element of \( \mathcal{G} \)).

So let \( a = h(\tilde{\delta}_{uvw}) \), \( b = h(\tilde{\delta}_{uvw}) \), and \( c = h(\tilde{\delta}_{uvw}) \). Observe that, for every \( x \in \{u,v\}^* \), \( h(\tilde{\delta}_{uvw}) = \tilde{\delta}_x \) \( R \), and so \( c \circ b \circ a = \text{id}_R \). Also, for any \( \tilde{\delta}_3(q) \in R \), \( a^2(\tilde{\delta}_3(q)) = (\tilde{\delta}_3)_R^2(\tilde{\delta}_3(q)) = \tilde{\delta}_{x_{uv}^2}(q) = \tilde{\delta}_x(q) \) by our assumption, so \( a^2 = \text{id}_R \).

On the other hand, \( q \in R \) as \( \tilde{\delta}_3(q) = q \), and \( \text{id}_R(q) = q \neq \tilde{\delta}_3(q) \) by assumption, so \( a \neq \text{id}_R \). As \( \sigma(a) \) divides \( 2 \), \( \sigma(a) = 2 \) follows. Similarly, we can show that \( \sigma(b) = k \) (using that \( \delta_{x_{uv}^k}(q) = \tilde{\delta}_x(q) \) for every \( x \in \{u,v\}^* \), and \( u \neq \tilde{\delta}_3(q) \)).

Finally, using that \( \delta_{x_{uv}^1}(q) = \tilde{\delta}_x(q) \) for every \( x \in \{u,v\}^* \), and \( u \neq \tilde{\delta}_3(q) \), we obtain that \( h(\tilde{\delta}_{uvw})^l = \text{id}_R \) and \( h(\tilde{\delta}_{uvw}) \neq \text{id}_R \). Therefore, it follows that \( \sigma(c) = \sigma(h(\tilde{\delta}_{uvw})) = \sigma(h(\tilde{\delta}_{uvw})) \neq \text{id}_R \), and so coprime to both \( 2 \) and \( k \), as required.

4 Deciding FO-definability: PSpace-hardness

Kozen [18] showed that deciding whether the intersection of the languages recognised by a set of given deterministic DFAs is non-empty is PSPACE-complete. By carefully analysing Kozen’s lower bound proof and using the criterion of Theorem 1 iii), Cho and Huynh [10] established that deciding FO(<)-definability of \( L(\mathcal{A}) \) is PSPACE-hard, for any given minimal DFA \( \mathcal{A} \). We generalise their construction and use the criteria in Theorem 1 (ii)–(iii) to cover FO(<,\( \equiv \)) and FO(<,\( \equiv \),\( \text{MOD} \))-definability as well.

**Theorem 2.** For any \( L \in \{\text{FO(<)}, \text{FO(<,\( \equiv \))}, \text{FO(<,\( \text{MOD} \))}\} \), deciding \( L \)-definability of the language \( L(\mathcal{A}) \) of a given minimal DFA \( \mathcal{A} \) is PSPACE-hard.

**Proof.** Let \( M \) be a deterministic Turing machine that decides a language using at most \( N = P_M(n) \) tape cells on any input of size \( n \), for some polynomial \( P_M \).

Given such an \( M \) and an input \( x \), our aim is to define three minimal DFAs whose languages are, respectively, \( \text{FO(<)}, \text{FO(<,\( \equiv \))}, \text{and FO(<,\( \text{MOD} \))}-\text{definable iff} \ M \) rejects \( x \), and whose sizes are polynomial in \( N \) and the size \( |M| \) of \( M \).

Suppose \( M = (Q,\Gamma,\gamma,b,q_0,q_{acc}) \) with a set \( Q \) of states, tape alphabet \( \Gamma \) with \( b \) for blank, transition function \( \gamma \), initial state \( q_0 \) and accepting state \( q_{acc} \).

Without loss of generality we assume that \( M \) erases the tape before accepting, its head is at the left-most cell in an accepting configuration, and if \( M \) does not accept the input, it runs forever. Given an input word \( x = x_1 \ldots x_n \) over \( \Gamma \), we represent configurations \( c \) of the computation of \( M \) on \( x \) by the \( N \)-long word written on the tape (with sufficiently many blanks at the end) in which the symbol \( y \) in the active cell is replaced by the pair \( (q,y) \) for the current state \( q \). The accepting computation of \( M \) on \( x \) is encoded by a word \( \bar{c}_1 \bar{c}_2 \ldots \bar{c}_k \) over the alphabet \( \Sigma = \Gamma \cup (Q \times \Gamma) \cup \{\bar{c},\bar{b}\}, \) with \( c_1, c_2, \ldots, c_k \) being the subsequent configurations. In particular, \( c_1 \) is the initial configuration on \( x \) (so it is of the form \( (q_0,x_1)x_2 \ldots x_n b \ldots b \)), and \( c_k \) is the accepting configuration (so it is of the
form \( (q_{\text{acc}}, b \ldots b) \). As usual for this representation of computations, we may regard \( \gamma \) as a partial function from \((\Gamma \cup (Q \times \Gamma) \cup \{2\})^3\) to \(\Gamma \cup (Q \times \Gamma)\) with \(\gamma(\sigma^j_{i-1}, \sigma^i_0, \sigma^j_{i+1}) = \sigma^j_{i+1}\) for each \(j < k\), where \(\sigma^j_i\) is the \(j\)th symbol of \(c^i\).

Let \( p_{M, x} = p \) be the first prime such that \( p \geq N + 2 \) and \( p \equiv \pm 1 \pmod{10} \). By [3] Corollary 1.6, \( p \) is polynomial in \( N \). Our first aim is to construct a \( p + 1 \)-long sequence \( \mathfrak{A}_i \) of disjoint minimal DFAs over \( \Sigma \). Each \( \mathfrak{A}_i \) has size polynomial in \( N \) and \( |M| \), and it checks certain properties of an accepting computation on \( x \) such that \( M \) accepts \( x \) iff the intersection of the \( L(\mathfrak{A}_i) \) is not empty and consists of the single word encoding the accepting computation on \( x \).

We define each \( \mathfrak{A}_i \) as an NFA, and assume that it can be turned to a DFA by adding a ‘trash state’ \( t_r \) looping on itself with every \( \sigma \in \Sigma \), and adding the missing transitions leading to \( t_r \). The DFA \( \mathfrak{A}_0 \) checks that an input starts with the initial configuration on \( x \) and ends with the accepting configuration:

When \( 1 \leq i \leq N \), the DFA \( \mathfrak{A}_i \) checks, for all \( j < k \), whether the \( j \)th symbol of \( c^j \) changes according to \( \gamma \) in passing to \( c^{j+1} \). The non-trash part of its transition function \( \delta^i \) is as follows, for \( 1 < i < N \). (For \( i = 1 \) and \( i = N \) some adjustments are needed.) For all \( u, u', v, w, w', y, z \in \Gamma \cup (Q \times \Gamma) \),

\[
\delta^i_{u}(t_r) = q^0, \quad \delta^i_u(q^j) = q^{j+1}, \quad \text{for } j = 0, \ldots, i-3, \quad \delta^i_u(q^{i-2}) = r_u, \quad \delta^i_v(r_u) = r_{uw},
\]

\[
\delta^i_w(r_{uv}) = q^0_{(u,v,w)}, \quad \delta^i_{u'}(q^j_{z^2}) = q^{j+1}, \quad \text{for } j = 0, \ldots, N-3, \quad j \neq N - i - 1,
\]

\[
\delta^i_z(q^j_{z^{N-i-1}}) = q^N_{z^{N-i-1}}, \quad \delta^i_{u'}(q^j_{z^{N-i-1}}) = f_i, \quad \delta^i_{w'}(q^j_{z^{N-2}}) = p_{u'z}, \quad \delta^i_{z}(p_{u'z}) = r_{u'z},
\]

see below, where \( z = \gamma(u, v, w) \) and \( z' = \gamma(u', z, w') \):

Finally, if \( N+1 \leq i \leq p \) then \( \mathfrak{A}_i \) accepts all words over \( \Sigma \) with a single occurrence of \( b \), which is the input’s last character.
Note that $A_{p-1} = A_p$ as $p \geq N + 2$. It is not hard to check that each of the $A_i$ is a minimal DFA that does not contain nontrivial cycles and the following holds:

**Lemma 1.** $M$ accepts $x$ iff $\bigcap_{i=0}^L L(A_i) \neq \emptyset$, in which case this language consists of a single word that encodes the accepting computation of $M$ on $x$.

Next, we require three sequences of DFAs $B^p_{\leq}$, $B^p_=$ and $B^p_{\text{MOD}}$, where $p > 5$ is a prime number with $p \neq \pm 1 \pmod{10}$; see the picture below for $p = 7$.

In general, the first sequence is $B^p_{\leq} = \{ \{ s_i \mid i < p \}, \{ a \}, \delta^p_{\leq}, s_0, \{ s_0 \} \}$, where $\delta^p_{\leq}(s_i) = s_j$ if $i, j < p$ and $j \equiv i + 1 \pmod{p}$. Then $L(B^p_{\leq})$ comprises all words of the form $(a^p)^*$, $B^p_{\leq}$ is the minimal DFA for $L(B^p_{\leq})$, and the syntactic monoid $M(B^p_{\leq})$ is the cyclic group of order $p$ (generated by the permutation $\delta^p_{\leq}$).

The second sequence is $B^p_\approx = \{ \{ s_i \mid i < p \}, \{ a, z \}, \delta^p_\approx, s_0, \{ s_0 \} \}$, where $\delta^p_\approx(s_i) = s_i$ and $\delta^p_\approx(s_i) = s_j$ if $i, j < p$ and $j \equiv i + 1 \pmod{p}$. One can check that $L(B^p_\approx)$ comprises all words of $a$'s and $z$'s where the number of $a$'s is divisible by $p$, $B^p_\approx$ is the minimal DFA for this language, and $M(B^p_\approx)$ is also the cyclic group of order $p$ (generated by the permutation $\delta^p_\approx$).

The third sequence is $B^p_{\text{MOD}} = \{ \{ s_i \mid i \leq p \}, \{ a, z \}, \delta^p_{\text{MOD}}, s_0, \{ s_0 \} \}$, where
- $\delta^p_{\text{MOD}}(s_p) = s_p$ and $\delta^p_{\text{MOD}}(s_i) = s_j$ whenever $i, j < p$ and $j \equiv i + 1 \pmod{p}$;
- $\delta^p_{\text{MOD}}(s_0) = s_p$, $\delta^p_{\text{MOD}}(s_p) = s_0$, and $\delta^p_{\text{MOD}}(s_i) = s_j$ whenever $1 \leq i, j < p$ and $i \cdot j \equiv p - 1 \pmod{p}$, that is, $j = -1/i$ in the finite field $\mathbb{F}_p$.

One can check that $B^p_{\text{MOD}}$ is the minimal DFA for its language, and the syntactic monoid $M(B^p_{\text{MOD}})$ is the permutation group generated by $\delta^p_{\leq}$ and $\delta^p_{\text{MOD}}$.

**Lemma 2.** For any prime $p > 5$ with $p \neq \pm 1 \pmod{10}$, the group $M(B^p_{\text{MOD}})$ is unsolvable, but all of its proper subgroups are solvable.

**Proof.** One can check that the order of the permutation $\delta^p_{\text{MOD}}$ is 2, that of $\delta^p_{\text{MOD}}$ is $p$, while the order of the inverse of $\delta^p_{\text{MOD}}$ is the same as the order of $\delta^p_{\text{MOD}}$, which is 3. So $M(B^p_{\text{MOD}})$ is unsolvable, for any prime $p$, by the Kaplan–Levy criterion. To prove that all proper subgroups of $M(B^p_{\text{MOD}})$ are solvable, we show that $M(B^p_{\text{MOD}})$ is a subgroup of the projective special linear group $\text{PSL}_2(p)$. If $p$ is a prime with $p > 5$ and $p \neq \pm 1 \pmod{10}$, then all proper subgroups of $\text{PSL}_2(p)$ are solvable; see, e.g., [17] Theorem 2.1. (So $M(B^p_{\text{MOD}})$ is in fact isomorphic to the unsolvable group $\text{PSL}_2(p)$.) Consider the set $P = \{0, 1, \ldots, p - 1, \infty\}$ of all points of the projective line over the field $\mathbb{F}_p$. By identifying $s_i$ with $i$ for $i < p$,
and \( s_p \) with \( \infty \), we may regard the elements of \( M(\mathcal{B}_p) \) as \( P \to P \) functions. The group \( \text{PSL}_2(p) \) consists of all \( P \to P \) functions of the form \( i \mapsto \frac{a_i + \zeta}{y + i} \), where \( w \cdot z - x \cdot y = 1 \), with the field arithmetic of \( \mathbb{F}_p \) extended by \( i + \infty = \infty \) for any \( i \in P \).  

One can check that the two generators of \( M(\mathcal{B}_p) \) are in \( \text{PSL}_2(p) \): take \( w = 1, x = 1, y = 0, z = 1 \) for \( \delta_a \), and \( w = 0, x = 1, y = p - 1, z = 0 \) for \( \delta_b \).

Finally, we define three automata \( \mathfrak{A}_<, \mathfrak{A}_=, \mathfrak{A}_\text{MOD} \) over the same tape alphabet \( \Sigma_+ = \Sigma \cup \{a_1, a_2, \ldots\} \), where \( a_1 \) and \( a_2 \) are fresh symbols. We take, respectively, \( \mathfrak{B}_< \), \( \mathfrak{B}_= \), \( \mathfrak{B}_\text{MOD} \), and replace each transition \( s_i \rightarrow a \) in them by a fresh copy of \( \mathfrak{A}_i \), for \( i \leq p \), as shown in the picture below.

We make \( \mathfrak{A}_<, \mathfrak{A}_=, \mathfrak{A}_\text{MOD} \) deterministic by adding a trash state \( tr \) looping on itself with every \( y \in \Sigma_+ \), and adding the missing transitions leading to \( tr \). It follows that \( \mathfrak{A}_<, \mathfrak{A}_=, \mathfrak{A}_\text{MOD} \) are minimal DFAs of size polynomial in \( N, |M| \).

\textbf{Lemma 3.} (i) \( L(\mathfrak{A}_<) \) is \( \text{FO}(<) \)-definable iff \( \bigcap_{i=0}^{\infty} L(\mathfrak{A}_i) = \emptyset \).

(ii) \( L(\mathfrak{A}_=) \) is \( \text{FO}(<, \equiv) \)-definable iff \( \bigcap_{i=0}^{\infty} L(\mathfrak{A}_i) = \emptyset \).

(iii) \( L(\mathfrak{A}_\text{MOD}) \) is \( \text{FO}(<, \text{MOD}) \)-definable iff \( \bigcap_{i=0}^{\infty} L(\mathfrak{A}_i) = \emptyset \).

\textbf{Proof.} As \( \mathfrak{A}_<, \mathfrak{A}_=, \mathfrak{A}_\text{MOD} \) are minimal, we can replace \( \sim \) by = in the conditions of Theorem 1. For the \( (\Rightarrow) \) directions, given some \( w \in \bigcap_{i=0}^{\infty} L(\mathfrak{A}_i) \), in each case we show how to satisfy the corresponding condition of Theorem 1: (i) take \( u = a_1 w_2, q = s_0 \), and \( k = p \); (ii) take \( u = a_1 w_2, v = \zeta^{|w|}, q = s_0 \), and \( k = p \); (iii) take \( u = \zeta, v = a_1 w_2, q = s_0 \), and \( k = p \).  

For the \( (\Leftarrow) \) directions, we show that the corresponding condition of Theorem 1 implies non-emptiness of \( \bigcap_{i=0}^{\infty} L(\mathfrak{A}_i) \). To this end, we define a \( \Sigma_+^* \rightarrow \{a, \zeta\}^* \) homomorphism by taking \( h(\zeta) = \zeta, h(a_1) = a \), and \( h(b) = \varepsilon \) for all other \( b \in \Sigma_+ \).

(i) and (ii): Let \( o \in \{<, \equiv\} \) and suppose \( q \) is a state in \( \mathfrak{A}_o \) and \( u' \in \Sigma_+^* \) such that \( q \neq \delta_{\mathfrak{A}_o}^u(q) \) and \( q = \delta_{\mathfrak{A}_o}^{a_2, k}(q) \) for some \( k \). Let \( S = \{s_0, s_1, \ldots, s_{p-1}\} \). We claim that there exist \( s \in S \) and \( u \in \Sigma_+^* \) such that

\begin{align*}
\delta_{\mathfrak{A}_o}^u(s) &\neq s, & (13) \\
\delta_{\mathfrak{A}_o}^u(s) &\in S, \quad \text{for every } x \in \{u\}^* & (14)
\end{align*}

Indeed, observe that none of the states along the cyclic \( q \rightarrow (u')^k \cdot q \) path \( \Pi \) in \( \mathfrak{A}_o \) is \( tr \). So there is some state along \( \Pi \) that is in \( S \), as otherwise one of the \( \mathfrak{A}_i \) would contain a nontrivial cycle. Therefore, \( u' \) must be of the form \( w_1 a_1 w' \) for some \( w \in \Sigma_+^* \), \( n < \omega \) and \( w' \in \Sigma_+^* \). It is easy to see that \( s = \delta_{\mathfrak{A}_o}^{a_2, k-1}(q) \) and \( u = \zeta^n a_1 w'w \) is as required in (13) and (14).
As \( M(\mathcal{B}_p) \) is a finite group, the set \( \{ \delta_{h(x)}^{p\mathcal{B}_p} \mid x \in \{u\}^* \} \) forms a subgroup \( \mathcal{G} \) in it (the subgroup generated by \( \delta_{h(x)}^{p\mathcal{B}_p} \)). We show that \( \mathcal{G} \) is nontrivial by finding a nontrivial homomorphic image of it. To this end, (13) implies that, for every \( x \in \{u\}^* \), the restriction \( \delta_x^{p\mathcal{B}_p} \) of \( \delta_x^{\mathcal{B}_p} \) to the set \( S = \{ \delta_y^{p\mathcal{B}_p} \mid y \in \{u\}^* \} \) is an \( S' \to S' \) function and \( \delta_x^{p\mathcal{B}_p} \) is a permutation of \( S' \), for every \( x \in \{u\}^* \). Thus, \( \{ \delta_{h(x)}^{p\mathcal{B}_p} \mid S' \mid x \in \{u\}^* \} \) is a homomorphic image of \( \mathcal{G} \) that is nontrivial by (13).

As \( \mathcal{G} \) is a nontrivial subgroup of the cyclic group \( M(\mathcal{B}_p) \) of order \( p \) and \( p \) is a prime, \( \mathcal{G} = M(\mathcal{B}_p) \). Then there is \( x \in \{u\}^* \) with \( \delta_{h(x)}^{p\mathcal{B}_p} = \delta_{h(x)}^{\mathcal{B}_p} \) (a permutation containing the \( p \)-cycle \((s_0s_1\ldots sp_{-1})\) ‘around’ all elements of \( S \)), and so \( S = S \) and \( x = z^na_1w_2w' \) for some \( n < \omega, x \in \Sigma^* \), and \( w' \in \Sigma_x^* \). As \( n = 0 \) when \( x = x \) and \( \delta_x^{p\mathcal{B}_p} \) is a nontrivial homomorphic image of it. To this end, (14) implies that, for every \( S = S \) and \( x \in \{u\}^* \), \( \delta_{h(x)}^{p\mathcal{B}_p} \) is a permutation of \( S' \), for every \( x \in \{u\}^* \). Thus, \( \{ \delta_{h(x)}^{p\mathcal{B}_p} \mid S' \mid x \in \{u\}^* \} \) is a homomorphic image of \( \mathcal{G} \) that is nontrivial by (13).

Indeed, by an argument similar to the one in the proof of (i) and (ii) above, we must have \( u' = w_u^na_1w_u' \) and \( v' = w_v^ma_1w_v' \) for some \( w_u, w_v \in \Sigma^* \), \( n < \omega \) and \( w_v' \in \Sigma_v^* \). For every \( x \in \{u,v\}^* \), as both \( \delta_{h(x)}^{p\mathcal{B}_p} \) and \( \delta_{h(x)}^{p\mathcal{B}_p} \) are in \( S' \), they must be the same state. Using this it is not hard to see that \( s = \delta_{h(x)}^{p\mathcal{B}_p} \), \( u = z^na_1w_u \) and \( v = z^ma_1w_v \) are as required in (15)–(17).

As \( M(\mathcal{B}_p^{\mathcal{M}^*}) \) is a finite group, the set \( \{ \delta_{h(x)}^{\mathcal{B}_p^{\mathcal{M}^*}} \mid x \in \{u,v\}^* \} \) forms a subgroup \( \mathcal{G} \) in it (the subgroup generated by \( \delta_{h(x)}^{p\mathcal{B}_p} \) and \( \delta_{h(x)}^{p\mathcal{B}_p} \)). We show that \( \mathcal{G} \) is unsolvable by finding an unsolvable homomorphic image of it. To this end, we let \( S' = \{ \delta_{h(x)}^{\mathcal{B}_p} \mid y \in \{u,v\}^* \} \). Then (10) implies that \( S' \subseteq S \) and

\[
\delta_{h(x)}^{p\mathcal{B}_p} (S') = \delta_{h(x)}^{p\mathcal{B}_p} (S') \subseteq S', \quad \text{for all } S' \in S \text{ and } x \in \{u,v\}^*,
\]

and so the restriction \( \delta_{h(x)}^{p\mathcal{B}_p} \mid S' \) of \( \delta_{h(x)}^{p\mathcal{B}_p} \) to \( S' \) is an \( S' \to S' \) function and \( \delta_{h(x)}^{p\mathcal{B}_p} \mid S' = \delta_{h(x)}^{p\mathcal{B}_p} \mid S' \). As \( M(\mathcal{B}_p^{\mathcal{M}^*}) \) is a group of permutations on a set containing \( S' \), \( \delta_{h(x)}^{p\mathcal{B}_p} \mid S' \) is a permutation of \( S' \), for any \( x \in \{u,v\}^* \). So \( \delta_{h(x)}^{p\mathcal{B}_p} \mid S' \mid x \in \{u,v\}^* \) is an unsolvable homomorphic image of \( \mathcal{G} \) that is unsolvable by the Kaplan–Levy criterion:

By (13), (14), and 2 and \( k \) being primes, the order of the permutation \( \delta_{h(x)}^{p\mathcal{B}_p} \mid S' \)
is 2, the order of $\delta^{\text{MOD}}_h$ is $k$, and the order of $\delta^{\text{MOD}}_h \upharpoonright S'$ is $k$, which is the same as the order of its inverse) is a $1$ divisor of $l$, and so coprime to both $2$ and $k$.

As $\mathcal{G}$ is an unsolvable subgroup of $M(\mathbb{Z}_k^{\text{MOD}})$, it follows from Lemma 2 that $\mathcal{G} = M(\mathbb{Z}_k^{\text{MOD}})$, and so $\{u, v\} \not\subseteq \mathcal{G}$. We claim that $S' = S$ also follows. Indeed, let $x \in \{u, v\}^*$ be such that $\delta^{\text{MOD}}_h(x) = \delta^{\text{MOD}}_h(x)$. As $|S'| \geq 2$ by (15), $s \in \{s_0, \ldots, s_{p-1}\}$ must hold, and so $\{s_0, \ldots, s_{p-1}\} \subseteq S'$ follows by (18). As there is $y \in \{u, v\}^*$ with $\delta^{\text{MOD}}_h(y) = \delta^{\text{MOD}}_h$, $s_p \in S'$ also follows by (18). Finally, as $\{u, v\} \not\subseteq \mathcal{G}$, there is $x \in \{u, v\}^*$ of the form $u^n a_1 a_2 w'$, for some $n < \omega$, $w \in \Sigma$ and $w' \in \Sigma^*_\omega$. As $S' = S$, $\delta^{\text{MOD}}_h(s_i) \in S$ for every $i \leq p$, and so $w \in \bigcap_{i=0}^p L(\mathfrak{A}_i)$.

Theorem 2 clearly follows from Lemmas 1 and 3.

5 Deciding $\mathcal{L}$-definability of 2NFAs in PSpace

Using the criterion Theorem 1 (i), Stern [25] showed that deciding whether the language of any given DFA is FO($<$)-definable can be done in PSPACE. In this section, we also use the criteria of Theorem 1 to provide PSPACE-algorithms deciding whether the language of any given 2NFA is $\mathcal{L}$-definable, whenever $\mathcal{L} \in \{\text{FO}(<), \text{FO}(<, \exists), \text{FO}(<, \text{MOD})\}$. Let $\mathfrak{A} = (Q, \Sigma, \delta, Q_0, F)$ be a 2NFA. Following [3], we first construct a (an exponential size) DFA $\mathfrak{A}'$ such that $L(\mathfrak{A}) = L(\mathfrak{A}')$. To this end, for any $w \in \Sigma^*$, we introduce four binary relations $b_{lr}(w), b_{rl}(w), b_{rr}(w)$, and $b_{ll}(w)$ on $Q$ describing the left-to-right, right-to-left, right-to-right, and left-to-left behaviour of $\mathfrak{A}$ on $w$. Namely,

- $(q, q') \in b_{lr}(w)$ if there is a run of $\mathfrak{A}$ on $w$ from $(q, 0)$ to $(q', w)$;
- $(q, q') \in b_{rl}(w)$ if there is a run of $\mathfrak{A}$ on $w$ from $(q, |w| - 1)$ to $(q', 0)$;
- $(q, q') \in b_{rr}(w)$ if, for some $a \in \Sigma$, there is a run on $aw$ from $(q, |aw| - 1)$ to $(q', 0)$ such that no $(q', 0)$ occurs in it before $(q', 0)$;
- $(q, q') \in b_{ll}(w)$ if, for some $a \in \Sigma$, there is a run on $aw$ from $(q, 1)$ to $(q', 0)$ such that no $(q', 0)$ occurs in it before $(q', 0)$.

For $w = \varepsilon$ (the empty word), we define the $b_{ij}(w)$ as the identity relation on $Q$. Let $b = (b_{lr}, b_{rl}, b_{rr}, b_{ll})$, where $b_{ij}$ are the behaviours of $\mathfrak{A}$ on some $w \in \Sigma^*$, in which case we can also write $b(w)$, and let $b' = b(w')$, for some $w' \in \Sigma^*$. We define the composition $b \cdot b' = b''$ with components $b''_{ij}$ as follows. Let $X$ and $Y$ be the transitive closure of $b_{ij} \circ b_{rl}$, and $b_{rr} \circ b'_{ij}$, respectively. Then we set:

$$
\begin{align*}
  b'_{lr} &= b_{lr} \circ b'_{ll} \cup b_{ll} \circ X \circ b'_{rr}, & b'_{rl} &= b'_{ll} \circ b_{rl} \cup b'_{rr} \circ Y \circ b_{rl}, \\
  b'_{rr} &= b_{rr} \cup b_{rl} \circ b'_{lr} \circ Y \circ b_{rl}, & b'_{ll} &= b_{ll} \cup b_{rl} \circ X \circ b'_{rr} \circ b_{rl}.
\end{align*}
$$

One can check that $b'' = b(w'w)$. Define a DFA $\mathfrak{A}' = (Q', \Sigma', q_0', F')$ by taking

$$
\begin{align*}
  Q' &= \{(B_{lr}, B_{rr}) \mid B_{lr} \subseteq Q_0 \times Q, B_{rr} \subseteq Q \times Q\}, & q_0' &= \{(q, q) \mid q \in Q_0\}, \\
  F' &= \{(B_{lr}, B_{rr}) \mid (q_0, q) \in B_{lr}, \text{ for some } q_0 \in Q_0 \text{ and } q \in F\}, \\
  \delta'_{a'}((B_{lr}, B_{rr})) &= (B'_{lr}, B'_{rr}), \text{ with } B'_{lr} = B_{lr} \circ X(a) \circ b_{lr}(a), \\
  B'_{rr} &= B_{rr} \cup b_{rl}(a) \circ Y(a) \circ b_{rr}(a),
\end{align*}
$$

$$
where $X(a)$ and $Y(a)$ are the reflexive and transitive closures of $b_{l}(a) \circ B_{r}$ and $B_{rr} \circ b_{r}(a)$, respectively. It is not hard to see that, for any $w \in \Sigma^{*}$,

$$\delta'_{u}((B_{r}, B_{rr})) = (B'_{r}, B'_{rr}) \text{ iff } B'_{r} = B_{r} \circ X(w) \circ b_{r}(w) \text{ and } B'_{rr} = B_{rr} \cup b_{r}(w) \circ B_{r} \circ Y(w) \circ b_{r}(w),$$

(19)

where $X(w)$ and $Y(w)$ are the reflexive and transitive closures of $b_{l}(w) \circ B_{r}$ and $B_{rr} \circ b_{r}(w)$, respectively. Also, one can show in a way similar to [22][29] that

$$L(\mathfrak{A}) = L(\mathfrak{A}') .$$

(20)

Next, we show that, even if the size of $\mathfrak{A}'$ is exponential in $\mathfrak{A}$, we can still use Theorem 1 to decide $\mathcal{L}$-definability of $L(\mathfrak{A})$ in $\text{PSpace}$:

**Theorem 3.** For $\mathcal{L} \in \{ \text{FO(<), FO(<, \equiv), FO(<, \text{MOD})} \}$, deciding $\mathcal{L}$-definability of $L(\mathfrak{A})$, for any 2NFA $\mathfrak{A}$, is in $\text{PSpace}$.

**Proof.** Let $\mathfrak{A}'$ be the DFA defined above for the given 2NFA $\mathfrak{A}$. By Theorem 1(i) and (20), $L(\mathfrak{A})$ is not FO(<)-defineable if there exist a word $w \in \Sigma^{*}$, a reachable state $q \in Q'$, and a number $k \leq |Q'|$ such that $q \not\sim_{q} \delta_{q}(q)$ and $q = \delta_{q}^{k}(q)$. We guess the required $k$ in binary, $q$ and a quadruple $b(u)$ of binary relations on $Q$. Clearly, they all can be stored in polynomial space in $|\mathfrak{A}|$. To check that our guesses are correct, we first check that $b(u)$ indeed corresponds to some $w \in \Sigma^{*}$. This is done by guessing a sequence $b_{0}, \ldots, b_{n}$ of distinct quadruples of binary relations on $Q$ such that $b_{0} = b(u_{0})$ and $b_{i+1} = b_{i} \cdot b(u_{i+1})$, for some $u_{0}, \ldots, u_{n} \in \Sigma$. (Any sequence with a subsequence starting after $b_{i}$ and ending with $b_{i+m}$, for some $i$ and $m$ such that $b_{i} = b_{i+m}$, is equivalent, in the context of this proof, to the sequence with such a subsequence removed.) Thus, we can assume that $n \leq 2^{|Q'|}$, and so $n$ can be guessed in binary and stored in $\text{PSpace}$. So, the stage of our algorithm checking that $b(u)$ corresponds to some $u \in \Sigma^{*}$ makes $n$ iterations and continues to the next stage if $b_{n} = b(u)$ or terminates with an answer no otherwise. Now, using $b(u)$, we compute $b(u^{k})$ by means of a sequence $b_{0}, \ldots, b_{k}$, where $b_{0} = b(u)$ and $b_{k+1} = b_{k} \cdot b(u_{k+1})$. With $b(u)$ ($b(u^{k})$), we compute $\delta'_{u}(q)$ (respectively, $\delta'_{u}(q)$) in $\text{PSpace}$ using (19). If $\delta'_{u}(q) \neq q$, the algorithm terminates with an answer no. Otherwise, in the final stage of the algorithm, we check that $\delta'_{u}(q) \neq q$. This is done by guessing $v \in \Sigma^{*}$ such that $\delta'_{u}(q) = q_{1}$, $\delta'_{v}(q) = q_{2}$, and $q_{1} \in F'$ if $q_{1} \not\in F'$. We guess such a $v$ (if exists) in the form of $b(v)$ using an algorithm analogous to that for guessing $u$ above.

By Theorem 1(ii) and (20), $L(\mathfrak{A})$ is not FO(<,$\equiv$)-defineable if there there exist words $u, v \in \Sigma^{*}$, a reachable state $q \in Q'$, and a number $k \leq |Q'|$ such that $q \not\sim_{q} \delta_{q}(q)$, $q = \delta_{q}^{1}(q)$, $|v| = |u|$, and $\delta_{v}^{i}(q) = \delta_{v}^{i}(q)$, for all $i \leq k$. We outline how to modify the algorithm for FO(<) above to check FO(<,$\equiv$)-defineability. First, we need to guess and check $v$ in the form of $b(v)$ in parallel with guessing and checking $u$ in the form of $b(u)$, making sure that $|v| = |u|$. For that, we guess a sequence of distinct pairs $(b_{0}, b'_{0}), \ldots, (b_{n}, b'_{n})$ such that the $b_{i}$ are as above, $b'_{0} = b(m)$ and $b'_{i+1} = b'_{i} \cdot b(v_{i+1})$, for some $v_{0}, \ldots, v_{n} \in \Sigma$. (Any such sequence with a subsequence starting after $(b_{i}, b'_{i})$ and ending with $(b_{i+m}, b'_{i+m})$, for some
It and $n$ such that $(b_i, b'_i) = (b_{i+m}, b'_{i+m})$, is equivalent to the sequence with that subsequence removed.) So $n \leq 2^{O(|Q|)}$. For each $i < k$, we can then compute $\delta'_u(q)$ and $\delta'_v(u, v, q)$, using (23), and check whether they are equal.

Finally, by Theorem 1 (iii) and (23), $L(\mathcal{A})$ is not $\text{FO}(\cdot, \text{MOD})$-definable if there exist $u, v \in \Sigma^*$, a reachable state $q \in Q'$ and $k, l \leq |Q'|$ such that $k$ is an odd prime, $l > 1$ and coprime to both 2 and $k$, $q \not\sim \delta'_s(q)$, $q \not\sim \delta'_u(q)$, $q \not\sim \delta'_v(q)$, and $\delta'_u(q) \sim \delta'_{zu}(q) \sim \delta'_{zu}(q) \sim \delta'_{zu}(q) \sim \delta'_{zu}(q)$, for all $x \in \{u, v\}^*$. We start by guessing $u, v \in \Sigma^*$ in the form of $b(u)$ and $b(v)$, respectively. Also, we guess $k$ and $l$ in binary and check that $k$ is an odd prime and $l$ is coprime to both 2 and $k$. By (19), $\delta'_x$ is determined by $b(x)$, for any $x \in \{u, v\}^*$. Thus, we can proceed as follows to verify that $u, v$ and $k$ and $l$ are as required. We perform the following steps, for each quadruple $b$ of binary relations on $Q$. First, we check whether $b = b(x)$, for some $x \in \{u, v\}^*$ (we discuss the algorithm for this below).

If this is not the case, we construct the next quadruple $b'$ and process it as this $b$. If it is the case, we compute all the states $\delta'_u(q)$, $\delta'_{zu}(q)$, $\delta'_{zu}(q)$, $\delta'_{zu}(q)$, $\delta'_u(q)$, $\delta'_v(q)$, and check their required (non)equivalences w.r.t. $\sim$, using the same method as for checking $\delta'_u(q) \not\sim q$ above. If they do not hold as required, our algorithm terminates with an answer no. Otherwise, we construct the next quadruple $b'$ and process it as this $b$. When all possible quadruples $b$ of binary relations of $Q$ have been processed, the algorithm terminates with an answer yes.

Now, to check that a given quadruple $b$ is equal to $b(x)$, for some $x \in \{u, v\}^*$, we simply guess a sequence $b_1, \ldots, b_n$ of quadruples of binary relations on $Q$ such that $b_0 = b(w_0)$, $b_n = b$ and $b_{i+1} = b_{i} \cdot b(w_{i+1})$, where $w_i \in \{u, v\}$. It follows from the argument above that it is enough to consider $n \leq 2^{O(|Q|)}$.

6 Further Research

The results obtained in this paper have been used for deciding the rewritability type of ontology-mediated queries (OMQs) given in linear temporal logic $\text{LTL}$ [22]. As mentioned in the introduction, $\text{LTL}$ OMQs can be simulated by automata. In the worst case, the automata are of exponential size, and deciding $\text{FO}$-rewritability of some OMQs may become $\text{ExpSpace}$-complete. On the other hand, there are natural and practically important fragments of $\text{LTL}$ with automata of special forms whose $\text{FO}$-rewritability can be decided in $\text{PSpace}$, $\Pi^p_2$ or $\text{coNP}$. However, it remains to be seen whether the corresponding algorithms, even in the simplest case of $\text{FO}(\cdot, \cdot)$-definability, are efficient enough for applications in temporal OBDA. Note that the problems considered in this paper are also relevant to the optimisation problem for recursive SQL queries.

Acknowledgements. This work was supported by UK EPSRC EP/S032282.

References

1. A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. Ontology-mediated query answering over temporal data: A survey. In S. Schewe, T. Schneider, J. Wijsen, eds., TIME 2017, vol. 90 of LIPIcs 1:1–1:37.
2. A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. First-order rewritability of ontology-mediated queries in linear temporal logic. *Artificial Intelligence*, vol. 299, October 2021.

3. D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC$^1$. *J. Comput. Syst. Sci.*, 38(1):150–164, 1989.

4. D. Barrington, K. Compton, H. Straubing, and D. Thérien. Regular languages in NC$^1$. *J. Comput. Syst. Sci.*, 44(3):478–499, 1992.

5. M. Beaudry, P. McKenzie, and D. Thérien. The membership problem in aperiodic transformation monoids. *J. ACM*, 39(3):599?616, 1992.

6. M. Bennett, G. Martin, K. O'Bryant, and A. Rechnitzer. Explicit bounds for primes in arithmetic progressions. *Illinois J. of Math.*, 62(1–4):427–532, 2018.

7. L. Bernátsky. Regular expression star-freeness is PSPACE-complete. *Acta Cybern.*, 13(1):1–21, 1997.

8. J.R. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6(1–6):66–92, 1960.

9. O. Carton and L. Dartois. Aperiodic Two-way Transducers and FO-Transductions. In S. Kreutzer, ed., CSL 2015, volume 41 of LIPIcs, pages 160–174, Dagstuhl.

10. S. Cho and D. Huynh. Finite-automaton aperiodicity is PSPACE-complete. *Theor. Comp. Sci.*, 88(1):99–116, 1991.

11. K. Compton and C. Laflamme. An algebra and a logic for NC$^1$. *Inf. Comput.*, 87(1/2):240–262, 1990.

12. C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98:21–51, 1961.

13. L. Fleischer and M. Kufleitner. The intersection problem for finite monoids. In R. Niedermeier and B. Vallée, eds., *STACS 2018*, volume 96 of LIPIcs, pages 1–14.

14. M. Holzer and B. König. Regular languages, sizes of syntactic monoids, graph colouring, state complexity results, and how these topics are related to each other. *Bull. EATCS*, 38:139–155, 2004.

15. S. Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

16. G. Kaplan and D. Levy. Solvability of finite groups via conditions on products of 2-elements and odd p-elements. *Bull. of Austr. Math. Soc.*, 82(2):265–273, 2010.

17. O. King. The subgroup structure of finite classical groups in terms of geometric configurations. In B. Webb, ed., *Surveys in Combinatorics*, vol. 327 of *London Math. Society Lecture Note Series*, pages 29–56. Cambridge University Press, 2005.

18. D. Kozen. Lower bounds for natural proof systems. In *Proc. of FOCS 1977*, pages 254–266. IEEE Computer Society Press, 1977.

19. R. McNaughton and S. Papert. *Counter-free automata*. The MIT Press, 1971.

20. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, 10:133–173, 2008.

21. J. Rotman. *An introduction to the theory of groups*. Springer-Verlag, 1999.

22. V. Ryzhikov, Y. Savateev, and M. Zakharyaschev. Deciding FO-rewritability of ontology-mediated queries in linear temporal logic. In C. Combi, J. Eder, M. Reynolds, eds., *TIME 2021*, LIPIcs 61:1–7:15.

23. M. Schützenberger. On finite monoids having only trivial subgroups. *Inf. Control.*, 8(2):190–194, 1965.

24. J. C. Shepherdson. The reduction of two-way automata to one-way automata. *IBM Journal of Research and Development*, 3(2):198–200, 1959.

25. J. Stern. Complexity of some problems from the theory of automata. *Inf. Control.*, 66(3):163–176, 1985.
26. H. Straubing. 

27. J. Thompson. Nonsolvable finite groups all of whose local subgroups are solvable. 

28. B. Trakhtenbrot. Finite automata and the logic of one-place predicates. 

29. M. Vardi. A note on the reduction of two-way automata to one-way automata. 

30. G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. Zakharyashev. Ontology-based data access: A survey. In J. Lang, ed., Proc. of IJCAI 2018., pages 5511–5519. ijcai.org, 2018.