View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Queen Mary Research Online

Bisimilarity in Fresh-Register Automata

Andrzej S. Murawski
University of Warwick
a.murawski@warwick.ac.uk

Steven J. Ramsay
University of Warwick
s.ramsay@warwick.ac.uk

Nikos Tzevelekos
Queen Mary University of London
nikos.tzevelekos@qmul.ac.uk

***Abstract*—Register automata are a basic model of computation over infinite alphabets. Fresh-register automata extend register automata with the capability to generate fresh symbols in order to model computational scenarios involving name creation. This paper investigates the complexity of the bisimilarity problem for classes of register and fresh-register automata. We examine all main disciplines that have appeared in the literature: general register assignments; assignments where duplicate register values are disallowed; and assignments without duplicates in which registers cannot be empty. In the general case, we show that the problem is EXPTIME-complete.**

However, the absence of duplicate values in registers enables us to identify inherent symmetries inside the associated bisimulation relations, which can be used to establish a polynomial bound on the depth of Attacker-winning strategies. Furthermore, they enable a highly succinct representation of the corresponding bisimulations. By exploiting results from group theory and computational group theory, we can then show solvability in PSPACE and NP respectively for the latter two register disciplines. In each case, we find that freshness does not affect the complexity class of the problem.

The results allow us to close a complexity gap for language equivalence of deterministic register automata. We show that deterministic language inequivalence for the no-duplicates fragment is NP-complete, which disproves an old conjecture of Sakamoto.

Finally, we discover that, unlike in the finite-alphabet case, the addition of pushdown store makes bisimilarity undecidable, even in the case of visibly pushdown storage.

***Index Terms*—register automata, bisimilarity, computational group theory, automata over infinite alphabets**

I. Introduction

Register automata are one of the simplest models of computation over infinite alphabets. They consist of finite-state control and finitely many registers for storing elements from the infinite alphabet. Since their introduction by Kaminski and Francez [14] as a candidate formalism for capturing regularity in the infinite-alphabet setting, they have been actively researched especially in the database and verification communities: selected applications include the study of markup languages [19] and run-time verification [11]. While register automata can detect symbols that are currently not stored in registers (local freshness), the bounded number of registers means that they are not in general capable of recognising inputs that are genuinely fresh in the sense that they occur in the computation for the first time (global freshness). Because such a feature is desirable in many contexts, notably dynamic resource allocation, the formalism has been extended in [27] to fresh-register automata, which do account for global freshness. This paper is concerned with the problem of *bisimilarity testing* for register and fresh-register automata.

Bisimulation is a fundamental notion of equivalence in computer science. Its central role is, in part, derived from the fact that it is intensional and yet very robust. Consequently, the algorithmics of bisimilarity have attracted a lot of attention from researchers interested in the theory and practice of equivalence checking. When the set of observable actions available to a system is finite, a lot is already known about the complexity of the problem for specific classes of systems, although tight bounds are often difficult to obtain in the infinite-state cases [26]. In this paper we prove a number of bounds on the complexity of bisimulation equivalence checking. We note that in this setting language equivalence is known to be undecidable [19].

Our results are expressed using a unified framework that comprises all variations that have appeared in the literature. They differ in the allowed register assignment discipline, which turns out to affect complexity. Assignments are allowed to be: (S) *single*, if the contents of all registers are required to be distinct; or (M) *multiple*, if we allow for duplicate values. Furthermore, registers are required to: (F) always be filled; or ($\#_0$) initially allowed to be empty; or ($\#$) allowed to be erased and filled during a run[1]. The complexity of bisimilarity checking for each combination is summarised in the table below, where we use the suffix "-c" to denote completeness for this class and "-s" to denote solvability only. The results hold regardless of whether one considers register or fresh-register automata.

$(M\#)$	$(M\#_0)$	(MF)	$(S\#)$	$(S\#_0)$	(SF)
EXP-c	EXP-c	EXP-c	EXP-c	PSPACE-c	NP-s

Our work thus provides a practical motivation for modelling systems with single assignment whenever possible — if the system does not need to erase the contents of registers mid-run, the corresponding equivalence problems are lower in the complexity hierarchy.

We start by giving coarse, exponential-time upper bounds for all the classes of system considered by showing how any such bisimilarity problem can be reduced to one for finite-state automata at exponential cost. For all the multiple assignment machines this bound is tight and, for single assignment, tightness depends upon whether or not erasing is allowed. The implied significance of being able to erase the contents of registers is explained by our proof that the bisimulation games associated with such systems can simulate the computations

[1]Empty content is "#". A full definition of each of the automaton variants is given in Section II.

of alternating Turing machines running in PSPACE. Here we set up an encoding of the tape, determined by the presence or absence of content in certain registers, and erasing of registers corresponds to writing of tape cells.

Once erasure is forbidden under single assignments, we obtain better bounds by investigating the structure of the associated bisimulation relations. Such relations are generally infinite, but only the relationship between the register assignments in two configurations is relevant to bisimilarity, and so we work with a finite, though exponentially large, class of symbolic relations built over partial permutations (to link register indices). Due to the inherent symmetry and transitivity of bisimilarity, each such relation forms an inverse semigroup under function composition. Also, crucially, the relations are upward closed in the information order. Although, taken separately, neither of the preceding facts leads to an exponential leap in succinctness of representation, taken together they reveal an interconnected system of (total) permutation groups underlying each relation. What is more, in any play of the associated bisimulation game, the number of registers that are empty must monotonically decrease. This, together with an application of Babai's result on the length of subgroup chains in symmetric groups [4], allows us to show that any violation of bisimilarity can be detected after polynomially many rounds of the bisimulation game. Consequently, in this case, we are able to decide bisimilarity in polynomial space.

The polynomial bound mentioned above enables us to close a complexity gap (between NP and PSPACE) in the study of deterministic language equivalence. Namely, we show that the language inequivalence problem for *deterministic* RA($S\#_0$) is solvable in NP, and thus NP-complete, refuting a conjecture by Sakamoto [21].

Further, if registers are additionally required to be filled (SF), we can exhibit very compact representations of the relevant bisimulation relations. The fact that permutation groups have small generating sets [16] allows us then to design a representation for symbolic bisimulations that is at most polynomial in size. Furthermore, by exploiting polynomial-time membership testing for permutation groups given in terms of their generators [10], we show that such a representation can be guessed and verified by a nondeterministic Turing machine in polynomial time.

Finally, we consider bisimilarity for visibly pushdown register automata (VPDRA) under the SF register discipline, and we show that the problem here is already undecidable. Since VPDRA(SF) are a particularly weak variant, this result implies undecidability for all PDRA considered in [18]. In contrast, for finite alphabets, bisimilarity of pushdown automata is known to be decidable [24] but non-elementary [5] and, in the visibly pushdown case, EXPTIME-complete [25].

Related Work. The complexity of bisimilarity problems has been studied extensively in the finite-alphabet setting and the current state of the art for infinite-state systems is summarised nicely in [26]. Recent papers concerning the complexity of decision problems for register automata have, until now, not considered bisimulation equivalence. However, there are several related complexity results in the concurrency literature.

In his PhD thesis, Pistore [20], gives an exponential-time algorithm for bisimilarity of HD-automata [17]. Since Pistore shows that bisimulation relations for HD-automata have many of the algebraic properties[2] as the relations we study here, it seems likely that our algorithm could be adapted to show NP-solvability of the bisimilarity problem for HD-automata. Indeed, a compact representation of symmetries using generators for such a purpose was envisaged by [8].

Jonsson and Parrow [13] and Boreale and Trevisan [7] consider bisimilarity over a class of data-independent processes. These processes are terms built over an infinite alphabet, but the behaviour of such a process does not depend upon the data from which it is built. In the latter work, the authors also consider a class of value-passing processes, whose behaviour may depend upon the result of comparing data for equality. They show that if such processes can be defined recursively then the problem is EXPTIME-complete. Since value passing can be seen as a purely functional proxy for multiple register assignments, this result neatly reflects our findings for RA($M\#$). Finally, decidability of bisimilarity for FRA($S\#_0$) was proven in [27], albeit without a proper study of its complexity (the procedure given in *loc. cit.* can be shown to run in NEXPTIME).

Finally, it would be interesting to see to what extent our decidability and complexity results can be generalised, e.g. in settings with ordered infinite alphabets or nominal automata [6].

Structure. In Section II we introduce the preliminaries and prove all of the EXPTIME bounds in Section III. Then we start the presentation of other results with register automata, as the addition of global freshness requires non-trivial modifications. In Section IV we show bounds for the ($S\#_0$) problems and apply the techniques to deterministic language equivalence in Section V. Section VI covers further improvements for the (SF) case. In Section VII we generalise our techniques to fresh-register automata and, finally, consider the pushdown case in Section VIII.

II. Preliminaries

We introduce some basic notation. Given a relation $R \subseteq X \times Y$, we define $\mathsf{dom}(R) = \{x \in X \mid \exists y.(x, y) \in R\}$ and $\mathsf{rng}(R) = \{y \in Y \mid \exists x.(x, y) \in R\}$. For natural numbers $i \leq j$, we write $[i, j]$ for the set $\{i, i + 1, \ldots, j\}$.

A. Bisimilarity

We define bisimulations generally with respect to a labelled transition system. As we shall see, the particular systems that we will be concerned with in this paper are the configuration graphs of various classes of (fresh-) register automata.

[2]E.g. the *active names* of [20] are comparable to our *characteristic sets*.

Definition 1. A ***labelled transition system*** (LTS) is a tuple $\mathcal{S} = (\mathbb{C}, \mathcal{A}ct, \{\xrightarrow{\ell} \mid \ell \in \mathcal{A}ct\})$, where $\mathbb{C}$ is a set of *configurations*, $\mathcal{A}ct$ is a set of *action labels*, and $\xrightarrow{\ell} \subseteq \mathbb{C} \times \mathbb{C}$ is a *transition relation* for each $\ell \in \mathcal{A}ct$.

A binary relation $R \subseteq \mathbb{C} \times \mathbb{C}$ is a ***bisimulation*** if for each $(\kappa_1, \kappa_2) \in R$ and each $\ell \in \mathcal{A}ct$, we have: (1) if $\kappa_1 \xrightarrow{\ell} \kappa_1'$, then there is some $\kappa_2 \xrightarrow{\ell} \kappa_2'$ with $(\kappa_1', \kappa_2') \in R$; (2) if $\kappa_2 \xrightarrow{\ell} \kappa_2'$, then there is some $\kappa_1 \xrightarrow{\ell} \kappa_1'$ with $(\kappa_1', \kappa_2') \in R$. We say that κ_1 and κ_2 are ***bisimilar***, written $\kappa_1 \sim \kappa_2$, just if there is some bisimulation R with $(\kappa_1, \kappa_2) \in R$.

Let us recall that bisimilarity has a very natural game-theoretic account. Given two configurations, one can consider a *bisimulation game* involving two players, traditionally called *Attacker* and *Defender* respectively. They play rounds in which Attacker fires a transition from one of the configurations and Defender has to follow with an identically labelled transition from the other configuration. In the first round, the chosen transitions must lead from the configurations to be tested for bisimilarity, while, in each subsequent round, they must start at the configurations reached after the preceding round. Defender loses if he cannot find a matching transition. In this framework, bisimilarity corresponds to the existence of a winning strategy for Defender. The process of playing a bisimulation game naturally favours Attacker as the decision maker but, thanks to the forcing technique of [12], it is possible to construct transition systems in which Defender effectively ends up making choices.

B. Fresh-register automata

We will be interested in testing bisimilarity of configurations generated by machines with registers and pushdown stack in the infinite-alphabet setting, i.e. as $\mathcal{A}ct$ we shall use the set $\Sigma \times \mathcal{D}$ for a finite alphabet Σ and an infinite alphabet $\mathcal{D}$ (with its elements sometimes called *names*), *cf.* data words [19].

Definition 2. Given a natural number r, a class of r-register assigments A is a set of functions from $[1, r]$ to $\mathcal{D} \uplus \{\#\}$. Fix such a class. An r-***fresh-register automaton*** (r-FRA) is a tuple $\mathcal{A} = \langle Q, q_0, \rho_0, \delta, F \rangle$, where:

- Q is a finite set of states, $q_0 \in Q$ initial and $F \subseteq Q$ final;
- $\rho_0 \in A$ is the initial r-register assignment;
- $\delta \subseteq Q \times \Sigma \times (\mathcal{P}([1, r]) \cup \{\circledast\}) \times [0, r] \times \mathcal{P}([1, r]) \times Q$ is the transition relation, with elements written as $q \xrightarrow{t, X, i, Z} q'$. We assume that in any such transition $i \notin Z$.

Finally an r-***register automaton*** (r-RA) is a special case of an r-FRA such that all its transitions $q \xrightarrow{t, X, i, Z} q'$ satisfy $X \neq \circledast$.

A register assignment then is just a mapping of register indices to letters from the infinite alphabet $\mathcal{D}$ and the special symbol $\#$. This symbol is used to represent the fact that a register is empty, i.e. contains no letter from $\mathcal{D}$. Consequently, by slight abuse of notation, for any r-register assignment ρ we will be writing $\mathsf{rng}(\rho)$ for the set $\rho([1, r]) \cap \mathcal{D}$, and $\mathsf{dom}(\rho)$ for $\rho^{-1}(\mathsf{rng}(\rho))$. Moreover, $\rho^{-1} = \{(d, i) \mid d \in \mathcal{D} \wedge (i, d) \in \rho\}$.

The meaning of a transition $q \xrightarrow{t, X, i, Z} q'$ is described as follows. The components t and X are a precondition: for the transition to be applicable, it must be that the next letter of the input has shape (t, a) for some $a \in \mathcal{D}$ and, moreover:

- if $X \subseteq [1, r]$ then a is already stored in exactly those registers named by X;
- if $X = \circledast$ then a is (globally) *fresh*: it has so far not appeared in the computation of $\mathcal{A}$.

If the transition applies then taking it results in changes being made to the current register assignment, namely: a is written into register i (unless $i = 0$, in which case it is not written at all) and all registers named by Z have their contents erased.

Definition 3. A ***configuration*** κ of an r-FRA $\mathcal{A}$ is a triple (q, ρ, H) consisting of a state $q \in Q$, an r-register assignment $\rho \in A$ and a finite set $H \subseteq \mathcal{D}$, called the *history*, such that $\mathsf{rng}(\rho) \subseteq H$. If $q_1 \xrightarrow{t, X, i, Z} q_2$ is a transition of $\mathcal{A}$, then a configuration (q_1, ρ_1, H_1) can make a transition to a configuration (q_2, ρ_2, H_2) accepting input (t, d), written $(q_1, \rho_1, H_1) \xrightarrow{(t, d)} (q_2, \rho_2, H_2)$, just if:

- $X = \{j \mid \rho_1(j) = d\}$, or $X = \circledast$ and $d \notin H$;
- for all $j \in [1, r]$, $\rho_2(j) = d$ if $j = i$; and $\rho_2(j) = \#$ if $j \in Z$; and $\rho_2(j) = \rho_1(j)$ otherwise;
- $H_2 = H_1 \cup \{d\}$.

We will sometimes write the set of configurations of $\mathcal{A}$ by $\mathbb{C}_{\mathcal{A}}$ and the induced transition relation by $\rightarrow_{\mathcal{A}}$. We let $\mathcal{S}(\mathcal{A})$ be the LTS $\langle \mathbb{C}_{\mathcal{A}}, \Sigma \times \mathcal{D}, \rightarrow_{\mathcal{A}} \rangle$.

On the other hand, a configuration κ of an r-RA $\mathcal{A}$ is a pair (q, ρ) of a state $q \in Q$ and an r-register assignment $\rho \in A$. The LTS $\langle \mathbb{C}_{\mathcal{A}}, \Sigma \times \mathcal{D}, \rightarrow_{\mathcal{A}} \rangle$ is defined precisely as above, albeit excluding the underlined conditions.

We define the specific classes of fresh-register automata that we will study in this work by considering specialisations of Definition 3 by the register assignment discipline followed.

Duplication in assignment. We consider two register storage policies, namely single assignment (S) or multiple assignment (M). In single assignment, we restrict the class of register assignments to be injective on non-empty registers, i.e. each $\rho \in A$ has, for all $i, j \in [1, r]$, $\rho(i) = \rho(j)$ just if $i = j$ or $\rho(i) = \# = \rho(j)$. In multiple assignment there is no such restriction. To ensure that all configurations respect the register assignment discipline, in an (S) automaton every transition $q_1 \xrightarrow{t, X, i, Z} q_2$ is required to have $X = \circledast$ or $X \subseteq \{i\}$.

Emptiness of registers. We consider the automaton's ability to process empty registers. We say that either all registers must always be filled (F), that registers may be initially empty ($\#_0$) or that the contents of registers may be erased ($\#$) during a run. Under condition (F), the class of r-register assignments A is restricted so that $\# \notin \rho([1, r])$ for each $\rho \in A$. Under conditions (F) and ($\#_0$), every transition $q_1 \xrightarrow{t, X, i, Z} q_2$ has $Z = \emptyset$ and $i \neq 0$. Condition ($\#$) imposes no specific restrictions.

We describe particular classes by the acronym FRA(XY) in which $X \in \{M, S\}$ and $Y \in \{F, \#_0, \#\}$. The class

FRA(XY) is the specialisation of Definition 2 to the largest class of register assignments A satisfying the constraints imposed by X and Y. E.g. FRA($S\#_0$) are those automata whose register assignments are all functions from $[1,r]$ to $\mathcal{D} \cup \{\#\}$ that are injective on non-empty registers, and every transition of such a machine is of the form $q_1 \xrightarrow{t,X,i,Z} q_2$ with $Z = \emptyset$, $i \neq 0$ and $X \in \{\circledast, \emptyset, \{i\}\}$. In a similar manner we define the classes RA(XY).

Remark 4. The class RA(MF) follow the register assignment discipline of the register automata defined by Segoufin [23]. The class RA($M\#_0$) follow the register assignment discipline of the M-Automata defined by Kaminski and Francez [14] and the class of RA($S\#_0$) follows the assignment discipline of the finite memory automata considered in the same paper. The class RA(SF) contain automata that follow the register assignment discipline of the machines considered by Nevin, Schwentick and Vianu [19]. The condition $i \neq 0$, which stipulates that every name encountered by the automaton be stored in some register, also originates from [14], [19]. The class FRA($S\#_0$) follow the register assignment discipline of the automata defined in [27].

In this paper we are concerned with the following family of decision problems.

Definition 5. Let $X \in \{M, S\}$ and $Y \in \{F, \#_0, \#\}$.

- The problem $\sim$-FRA(XY) is: given an FRA(XY) $\mathcal{A}$ and configurations $\kappa_1 = (q_1, \rho_1, H)$ and $\kappa_2 = (q_2, \rho_2, H)$, does $\kappa_1 \sim \kappa_2$ hold in $\mathcal{S}(\mathcal{A})$?
- The problem $\sim$-RA(XY) is: given an RA(XY) $\mathcal{A}$ and configurations κ_1 and κ_2 , does $\kappa_1 \sim \kappa_2$ hold in $\mathcal{S}(\mathcal{A})$?

We shall relate the various classes of bisimilarity problems that we study by their complexity. We write $P_1 \leq P_2$ to denote that there is a polynomial-time many-one reduction from problem P_1 to problem P_2.

Lemma 6. *The considered bisimilarity problems can be related as in Figure 1.*

C. Groups and permutations

For any $S \subseteq [1,n]$, we shall write $\mathcal{S}_S$ for the group of permutations on S, and $\mathcal{IS}_S$ for the inverse semigroup of partial permutations on S. For economy, we write $\mathcal{S}_n$ for $\mathcal{S}_{[1,n]}$; and $\mathcal{IS}_n$ for $\mathcal{IS}_{[1,n]}$.

For partial permutations σ and τ, we write $\sigma;\tau$ for their relational composition: $\sigma;\tau = \{\, (i,j) \mid \exists k.\sigma(i) = k \wedge \tau(k) = j \,\}$. Moreover, for any σ and $i, j \in [1,n]$, we let $\sigma[i \mapsto j]$ be the result of updating σ with (i,j):

$$\sigma[i \mapsto j] \;=\; \{(i,j)\} \cup \{(k,\sigma(k)) \mid k \neq i \wedge \sigma(k) \neq j\}.$$

For all $j \in [1,n]$, $\sigma \in \mathcal{S}_S$ and $S \subseteq [1,n]$ we also write $S[j]$ for $S \cup \{j\}$, and $\sigma \cdot S$ for $\{\sigma(i) \mid i \in S\}$.

III. Bisimilarity problems complete for EXPTIME

In this section we show that the upper four classes in our two hierachies of automata all have bisimilarity problems that are complete for exponential time.

Theorem 7. *All of the problems $\sim$-RA($S\#$), $\sim$-RA(MF), $\sim$-RA($M\#_0$), $\sim$-RA($M\#$), $\sim$-FRA($S\#$), $\sim$-FRA(MF), $\sim$-FRA($M\#_0$) and $\sim$-FRA($M\#$) are EXPTIME-complete.*

Proof: The result follows immediately from Propositions 8 and 10 and Lemma 6. ∎

Our argument proceeds by showing that $\sim$-FRA($M\#$) can be solved in EXPTIME (Proposition 8) and $\sim$-RA($S\#$) is already EXPTIME-hard (Proposition 10). For the former, we reduce the problem to a bisimilarity problem for finite state automata of exponential size.

Given an instance of the r-register, FRA($M\#$) bisimilarity problem, the idea is to construct a bisimilarity problem for a finite automaton over an alphabet derived from a finite subset $N \subseteq \mathcal{D}$ of size $2r+2$. Given a configuration $\kappa = (q, \rho, H)$ of the FRA, we represent it by an abstract configuration $\phi \cdot \kappa = (q, \phi \cdot \rho, \phi \cdot H)$ which is built entirely from letters in N. Here $\phi : \mathcal{D} \to N$ is surjective, $\phi \cdot \rho = (\phi[\# \mapsto \#]) \circ \rho$ and $\phi \cdot H = \{\phi(d) \mid d \in H\}$. We choose the abstraction ϕ in such a way that it partitions $\mathcal{D}$ and N with respect to $\mathsf{rng}(\rho)$ and H: that is, $\phi = \phi_1 \uplus \phi_2 \uplus \phi_3$ where $\mathsf{rng}(\phi_i)$ are all distinct and $\mathsf{dom}(\phi_1) = \mathsf{rng}(\rho)$, $\mathsf{dom}(\phi_2) = H \setminus \mathsf{rng}(\rho)$ and $\mathsf{dom}(\phi_3) = \mathcal{D} \setminus H$. In addition, ϕ_1 is injective.

The partitioning conditions ensure that our representation by abstract configurations is faithful. But, due to global freshness, the abstraction ϕ cannot be chosen uniformly for the entire simulation. This is because, with the alphabet limited to N, there would be no letters available to be played as part of globally fresh transitions as soon as the simulated history $\phi \cdot H$ became equal to N. Hence, the simulation needs to recycle letters in the history as soon as they become otherwise irrelevant to the current configuration and, consequently, a new (typically smaller) history and a new abstraction ϕ' must be chosen at each step. However, at position $(q_1, \phi \cdot \rho_1, \phi \cdot H), (q_2, \phi \cdot \rho_2, \phi \cdot H)$ of the simulation, the only letters that can be recycled are those that are not in $\phi \cdot \rho_1$ or $\phi \cdot \rho_2$. Recycling such a letter d by removing it from $\phi \cdot H$ is unfaithful to the simulation, since it would potentially allow a globally fresh transition playing d to be matched by a local one. This demonstrates that it is necessary to know *both* register assignments of the position in order to choose which letters are available to recycle and hence the shape of a new history.

To this end, the bisimulation game induced by the simulating finite automaton is constructed so that both of the two component systems contain both of $\phi \cdot \rho_1$ and $\phi \cdot \rho_2$.

Proposition 8. *$\sim$-FRA($M\#$) is solvable in EXPTIME.*

Proof sketch: Given an r-FRA($M\#$) $\mathcal{A} = \langle Q, q_0, \rho_0, \delta, F \rangle$ and a pair of input configurations, we decide their bisimilarity by checking an equivalent bisimilarity problem on a finite-state automaton $\mathcal{B}$. Each state of the finite automaton is of the form $(\sigma, q_1, \rho_1, H, q_2, \rho_2, p)$, where ρ_1 and ρ_2 are r-register assignments drawn only from N, representing the left and right component systems of the bisimulation game that is being simulated. States q_1 and q_2 are from Q, H is a history built only over N and σ and p are bookeeping information

$$\begin{array}{ccccccccccc}
\sim\text{-FRA}(SF) & \leq & \sim\text{-FRA}(S\#_0) & \leq & \sim\text{-FRA}(S\#) & \leq & \sim\text{-FRA}(MF) & \leq & \sim\text{-FRA}(M\#_0) & \leq & \sim\text{-FRA}(M\#) \\
\vee\!| & & \vee\!| & & \vee\!| & & \vee\!| & & \vee\!| & & \vee\!| \\
\sim\text{-RA}(SF) & \leq & \sim\text{-RA}(S\#_0) & \leq & \sim\text{-RA}(S\#) & \leq & \sim\text{-RA}(MF) & \leq & \sim\text{-RA}(M\#_0) & \leq & \sim\text{-RA}(M\#)
\end{array}$$

Fig. 1. Relationship between the main bisimilarity problems considered in this work.

drawn from sets of constant size. The construction induces a bisimulation game in which one turn of the original game is simulated in two parts, consisting of four turns.

In the first part, Attacker announces which component he would like to play from ($i \in \{1, 2\}$) and then which transition $T \in \delta$ he would like to simulate and which letter from N to do it with. He then updates register assignment ρ_i in the state accordingly, leaving ρ_{3-i} untouched. Defender responds by announcing the same transition and letter and updating the same register assignment ρ_i but in the other component.

In the second part, Defender uses Defender forcing in order to choose which transition she would like to simulate in response to Attacker's choice T, the letter used to simulate it (which must be the same as the one chosen by Attacker) and what the value of the new history should be. She then updates register assignment ρ_{3-i} and the history in her component accordingly. Attacker is forced to respond by announcing the same transition, letter and choice of history and updating assignment ρ_{3-i} and the history in his component to match.

It can be shown that such a simulation gives a faithful reduction from the FRA bisimilarity problem to the bisimilarity problem for finite automata. The number of states of the automata is bounded by $O(|Q| \cdot 2^{|N|+2r\log|N|})$. The alphabet of the finite automaton, whose letters simultaneously announce a transition of the FRA, a letter from N and a history, are bounded above by $O(|\delta| \cdot 2^{|N|})$. Hence bisimilarity can be solved in time $O(|\delta| \cdot |Q|^2 \cdot 2^{3|N|+4r|N|\log|N|})$. Since $|N|$ is $O(r)$, it follows that bisimilarity can be decided in time exponential in the size of the FRA, or polynomial in the size of the FRA for fixed r. ∎

Remark 9. The preceding proof shows that one turn of the FRA($M\#$) bisimulation game can be simulated by using four turns of the bisimulation game for the simulating finite automaton. Consequently, any winning strategy for the FRA-induced game can be transformed into a winning strategy for the finite automaton-induced game with at most a constant-factor increase in depth.

Further down the hierachy, to show $\sim$-RA($S\#$) is EXPTIME-hard, we use the registers of this class of automata to represent the tape content of bounded Turing machines.

Proposition 10. *$\sim$-RA($S\#$) is EXPTIME-hard.*

Proof sketch: We reduce instances of the Alternating Linear Bounded Automaton (ALBA) acceptance problem, which is known to be EXPTIME-hard, to RA($S\#$) bisimilarity. From an ALBA $\mathcal{M}$ we construct an RA($S\#$) $\mathcal{A}$ that simulates it, with the the binary tape content of $\mathcal{M}$ encoded by the register assignment of $\mathcal{A}$. At every step of the bisimulation game, we arrange for Defender to choose transitions from existential states (using Defender forcing [12]) and Attacker to choose from universal states.

The ability of $\mathcal{A}$ to use empty registers and to erase full registers is key to encoding the exponential amount of information held on the tape in a number of registers that is polynomial in its size. To this end, to represent a tape of size n, we equip $\mathcal{A}$ with $2n$ registers, under the following encoding. Cell k of the tape has 0 written on it iff register $2k-1$ is empty and it has 1 written on it iff register $2k$ is empty. ∎

IV. PSPACE-completeness for RAs with single assignment without erasure (RA($S\#_0$))

We next prove that the EXPTIME bound can be improved if duplicate values and erasures are forbidden. We handle register automata first to expose the flavour of our technique. The main result is given below, it follows from Propositions 21 and 22.

Theorem 11. *$\sim$-RA($S\#_0$) is PSPACE-complete.*

Simplified notation: Recall that, in any transition $q_1 \xrightarrow{t,X,i,Z} q_2$ of an r-RA($S\#_0$), we have that $Z = \emptyset$, $i \neq 0$ and $X \subseteq \{i\}$. These restrictions allow for a simpler notation for transitions, with $\delta \subseteq Q \times \Sigma \times ([1,r] \cup \{ i^\bullet \mid i \in [1,r] \}) \times Q$:

(a) we write each transition $q_1 \xrightarrow{t,\{i\},i,\emptyset} q_2$ as $q_1 \xrightarrow{t,i} q_2$;
(b) and each transition $q_1 \xrightarrow{t,\emptyset,i,\emptyset} q_2$ as $q_1 \xrightarrow{t,i^\bullet} q_2$.

Thus, transitions of type (a) correspond to the automaton reading an input (t, a) where a is the name in the i-th register; while in (b) transitions the automaton reads (t, a) if a is *locally fresh*, that is, it does not appear in the registers, and in this case a will be stored in register i.

A. Symbolic bisimulation

We attack the bisimulation problem *symbolically*, i.e. by abstracting actual names in the bisimulation game to the indices of the registers where these names reside. This will lead us to consider groups of finite permutations and inverse semigroups of partial finite permutations. We shall define symbolic bisimulations over pairs (q, S) of a state q and a set of register indices $S \subseteq [1, r]$. In this way, the locations of the empty registers $[1, r] \setminus S$ are made explicit.

Definition 12. Let $\mathcal{A} = \langle Q, q_0, \rho_0, \delta, F \rangle$ be an r-RA($S\#_0$). We first set:

$$\begin{aligned}
\mathcal{U}_0 &= Q \times \mathcal{P}([1,r]) \times \mathcal{IS}_r \times Q \times \mathcal{P}([1,r]) \\
\mathcal{U} &= \{ (q_1, S_1, \sigma, q_2, S_2) \in \mathcal{U}_0 \mid \sigma \subseteq S_1 \times S_2 \}
\end{aligned}$$

A *symbolic simulation* on $\mathcal{A}$ is a relation $R \subseteq \mathcal{U}$, with elements $(q_1, S_1, \sigma, q_2, S_2) \in R$ written infix $(q_1, S_1)\, R_\sigma\, (q_2, S_2)$, such

that all $(q_1, S_1, \sigma, q_2, S_2)$ satisfy the following *symbolic simulation conditions* (SYS)[3]:

- for all $q_1 \xrightarrow{t,i} q_1'$,
  - if $i \in \mathsf{dom}(\sigma)$ then there is some $q_2 \xrightarrow{t,\sigma(i)} q_2'$ with $(q_1', S_1)\, R_\sigma\, (q_2', S_2)$,
  - if $i \in S_1 \setminus \mathsf{dom}(\sigma)$ then there is some $q_2 \xrightarrow{t,j^\bullet} q_2'$ with $(q_1', S_1)\, R_{\sigma[i\mapsto j]}\, (q_2', S_2[j])$;
- for all $q_1 \xrightarrow{t,i^\bullet} q_1'$,
  - there is some $q_2 \xrightarrow{t,j^\bullet} q_2'$ with $(q_1', S_1[i])\, R_{\sigma[i\mapsto j]}\, (q_2', S_2[j])$,
  - for all $j \in S_2 \setminus \mathsf{rng}(\sigma)$, there is some $q_2 \xrightarrow{t,j} q_2'$ with $(q_1', S_1[i])\, R_{\sigma[i\mapsto j]}\, (q_2', S_2)$.

We let the inverse of R be

$$R^{-1} = \{\, (q_2, S_2, \sigma^{-1}, q_1, S_1) \mid (q_1, S_1, \sigma, q_2, S_2) \in R \,\}$$

and call R a ***symbolic bisimulation*** if both R and R^{-1} are symbolic simulations. We let *s-bisimilarity*, denoted $\overset{\mathsf{s}}{\sim}$, be the union of all symbolic bisimulations. We say that configurations (q_1, ρ_1) and (q_2, ρ_2) are *s-bisimilar*, written $(q_1, \rho_1) \overset{\mathsf{s}}{\sim} (q_2, \rho_2)$, if $(q_1, \mathsf{dom}(\rho_1)) \overset{\mathsf{s}}{\sim}_{\rho_1;\rho_2^{-1}} (q_2, \mathsf{dom}(\rho_2))$.

We approximate symbolic bisimilarity by a sequence of ***indexed bisimilarity*** relations $\overset{i}{\sim} \subseteq \mathcal{U}$ defined inductively as follows. First, we let $\overset{0}{\sim}$ be the whole of $\mathcal{U}$. Then, for all $i \in \omega$, $(q_1, S_1) \overset{i+1}{\sim}_\tau (q_2, S_2)$ just if $(q_1, S_1, \tau, q_2, S_2)$ and $(q_2, S_2, \tau^{-1}, q_1, S_1)$ both satisfy the (SYS) conditions in $\overset{i}{\sim}$.

Lemma 13. *Let (q_1, ρ_1), (q_2, ρ_2) be configurations of an r-RA($S\#_0$), then: $(q_1, \rho_1) \sim (q_2, \rho_2) \iff (q_1, \rho_1) \overset{\mathsf{s}}{\sim} (q_2, \rho_2)$. Furthermore, for all $i \in \omega$, $\overset{i+1}{\sim} \subseteq \overset{i}{\sim}$ and $(\bigcap_{i\in\omega} \overset{i}{\sim}) = \overset{\mathsf{s}}{\sim}$.*

Our next aim is to show that $\overset{\mathsf{s}}{\sim}$ and each $\overset{i}{\sim}$ are closed under composition and extension of partial permutations. The latter allows us, in Lemma 18, to bound the convergence of the indexed bisimulations by finding within them strict chains of subgroups. The former, in Section VI, helps us to represent $\overset{\mathsf{s}}{\sim}$ succinctly by appropriate choices of representatives.

Given $S_1, S_2 \subseteq [1, r]$ and $\sigma, \sigma' \in \mathcal{IS}_r$ we write $\sigma \leq_{S_1,S_2} \sigma'$ just if $\sigma \subseteq \sigma' \subseteq S_1 \times S_2$. Moroever, given $X \subseteq S \subseteq [1, r]$, we write id_X for the partial map from S to S that acts as identity on X (and is undefined otherwise).noteChanged the wording here as it wasn't clear what the range of S was (it looked like it captured the (ID) rule) For any $R \subseteq \mathcal{U}$, we define its ***closure*** $Cl(R)$ to be the smallest relation R' containing R and closed under the following rules.

$$\frac{}{(q, S, \mathsf{id}_S, q, S) \in R'}\ (\textsc{Id}) \qquad \frac{(q_1, S_1, \sigma, q_2, S_2) \in R'}{(q_2, S_2, \sigma^{-1}, q_1, S_1) \in R'}\ (\textsc{Sym})$$

$$\frac{(q_1, S_1, \sigma, q_2, S_2) \in R' \qquad \sigma \leq_{S_1,S_2} \sigma'}{(q_1, S_1, \sigma', q_2, S_2) \in R'}\ (\textsc{Ext})$$

$$\frac{(q_1, S_1, \sigma_1, q_2, S_2) \in R' \qquad (q_2, S_2, \sigma_2, q_3, S_3) \in R'}{(q_1, S_1, \sigma_1; \sigma_2, q_3, S_3) \in R'}\ (\textsc{Tr})$$

[3]We say that $(q_1, S_1, \sigma, q_2, S_2)$ *satisfies the* (SYS) *conditions in* R.

We say R is *closed* in case $Cl(R) = R$. We can show:

Lemma 14. *Let $P, R \subseteq \mathcal{U}$ be such that $R = R^{-1}$. If all $g \in R$ satisfy the* (SYS) *conditions in P then all $g \in Cl(R)$ satisfy the* (SYS) *conditions in $Cl(P)$.*

Much of the following development relies upon the fact that bisimilarity and indexed bisimilarity have a closed structure.

Corollary 15. *(Closures) Bisimilarity and indexed bisimilarity for RA($S\#_0$) are both closed:*
1) $\overset{\mathsf{s}}{\sim} = Cl(\overset{\mathsf{s}}{\sim})$; 2) for all $i \in \omega$: $\overset{i}{\sim} = Cl(\overset{i}{\sim})$.

Proof: For 1 note that $\overset{\mathsf{s}}{\sim} = (\overset{\mathsf{s}}{\sim})^{-1}$ and all its elements satisfy the (SYS) conditions in $\overset{\mathsf{s}}{\sim}$. Hence, by Lemma 14 we have that $Cl(\overset{\mathsf{s}}{\sim})$ is a symbolic bisimulation, i.e. $Cl(\overset{\mathsf{s}}{\sim}) = \overset{\mathsf{s}}{\sim}$. The result then follows. For 2 we proceed by induction on i. When $i = 0$ then the result follows from the fact that $\overset{0}{\sim}$ is the universal relation. For the inductive case, note first that $\overset{i+1}{\sim}$ is symmetric by construction and all $g \in \overset{i+1}{\sim}$ satisfy the (SYS) conditions in $\overset{i}{\sim}$. Hence, by Lemma 14, all elements of $Cl(\overset{i+1}{\sim})$ satisfy the (SYS) conditions in $Cl(\overset{i}{\sim})$. By IH, $Cl(\overset{i}{\sim}) = \overset{i}{\sim}$ so $Cl(\overset{i+1}{\sim}) \subseteq \overset{i+1}{\sim}$, as required. ∎

B. Permutation groups

Next we present a series of results that uncover group-theoretic structure in closed relations. Given $p \in Q$, $S \subseteq [1, r]$ and R closed, let $\mathcal{J}_S^p(R) = \{X \mid X \subseteq S,\ (p, S)\ R_{\mathsf{id}_X}\ (p, S)\}$.

Lemma 16. *$\mathcal{J}_S^p(R) \neq \emptyset$ and if $X_1, X_2 \in \mathcal{J}_S^p(R)$ then $X_1 \cap X_2 \in \mathcal{J}_S^p(R)$.*

Proof: $\mathcal{J}_S^p(R) \neq \emptyset$ follows from $S \in \mathcal{J}_S^p(R)$. For the rest, we observe that $\mathsf{id}_{X_1}; \mathsf{id}_{X_2} = \mathsf{id}_{X_1 \cap X_2}$ and R is closed. ∎

It follows from the lemma above that $\mathcal{J}_S^p(R)$ contains the least element with respect to inclusion, which we shall call *the **characteristic set** of (p, S) in R* and denote by $X_S^p(R)$. By Corollary 15, $\mathcal{J}_S^p(R) = \{X \mid X_S^p(R) \subseteq X \subseteq S\}$.

The family $\{X_S^p(R)\}_{p\in Q}$ turns out to play an important structural role in R for the following reason.

Lemma 17. *Let $p \in Q$ and $\mathcal{G}_S^p(R) = \{\sigma \cap (X_S^p(R) \times X_S^p(R)) \mid (p, S)\ R_\sigma\ (p, S)\}$. Then $\mathcal{G}_S^p(R)$ is a group (under composition). In particular, it is a subgroup of $\mathcal{S}_{X_S^p(R)}$.*

Proof (sketch): First, since $(p, S)\ R_{\mathsf{id}_S}\ (p, S)$, we have $\mathsf{id}_{X_S^p(R)} \in \mathcal{G}_S^p(R)$. Now, $(p, S)\ R_\sigma\ (p, S)$ implies $(p, S)\ R_{\sigma^{-1}}\ (p, S)$. The existence of inverses is proved by establishing that $\sigma \cap (X_S^p(R) \times X_S^p(R))$ and $\sigma^{-1} \cap (X_S^p(R) \times X_S^p(R))$ are bijective. Thus, $(\sigma \cap (X_S^p(R) \times X_S^p(R)))\,;(\sigma^{-1} \cap (X_S^p(R) \times X_S^p(R))) = \mathsf{id}_{X_S^p(R)}$. ∎

Since indexed bisimulations are closed, they have group-theoretic structure. We use it to help estimate their rate of convergence. Recall $\mathcal{U} = Q \times \mathcal{P}([1, r]) \times \mathcal{IS}_r \times Q \times \mathcal{P}([1, r])$.

Lemma 18. *Let $S_1, S_2 \subseteq [1, r]$ and $\mathcal{U}_{S_1,S_2} = Q \times \{S_1, S_2\} \times \mathcal{IS}_r \times Q \times \{S_1, S_2\}$. Then the sub-chain $\{\overset{i}{\sim} \mid (\overset{i+1}{\sim} \cap\, \mathcal{U}_{S_1,S_2}) \subsetneq (\overset{i}{\sim} \cap\, \mathcal{U}_{S_1,S_2})\}$ has size $O(|Q|^2 + r^2|Q|)$.*

Proof (sketch): We show that changes in $\overset{j}{\sim} \cap \mathcal{U}_{S_1,S_2}$ (as j increases) can be traced back to either shrinkage of a characteristic set $X^p_S(\overset{j}{\sim})$ ($S \in \{S_1, S_2\}$), or shrinkage of $\mathcal{G}^p_S(\overset{j}{\sim})$ ($S \in \{S_1, S_2\}$) or disappearance of all tuples $(q_1, S'_1, \sigma, q_2, S'_2)$ for some $q_1, q_2 \in Q$ and $S'_1, S'_2 \in \{S_1, S_2\}$. The number of changes of each kind can be bounded by a polynomial. In the second case we appeal to the fact that strict chains of subgroups of a symmetric group on n-elements have length at most linear in n, which is a result of Babai [4]. ∎

Note that it does not quite follow from the above result that the sequence $(\overset{i}{\sim})$ converges in polynomially many steps, because there are exponentially many pairs (S_1, S_2). Next we shall establish such a bound by studying more closely the overlap in evolutions of different (S_1, S_2). Let us write $\gamma(S_1, S_2)$ for $|S_1| + |S_2|$, i.e. $0 \leq \gamma(S_1, S_2) \leq 2r$.

Lemma 19. *Let $\mathcal{U}^-_{S_1,S_2} = Q \times \{S_1\} \times \mathcal{IS}_r \times Q \times \{S_2\}$ and let c be the constant of $O(|Q|^2 + r^2|Q|)$ in Lemma 18 (2).*

1) *Then, for any (S_1, S_2), we have $\overset{j}{\sim} \cap \mathcal{U}^-_{S_1,S_2} = \overset{\mathsf{s}}{\sim} \cap \mathcal{U}^-_{S_1,S_2}$, where $j = c(2r - \gamma(S_1, S_2) + 1)(|Q|^2 + r^2|Q|)$.*
2) *Let $B = c(2r + 1)(|Q|^2 + r^2|Q|)$. For any (S_1, S_2), $\overset{B}{\sim} \cap \mathcal{U}^-_{S_1,S_2} = \overset{\mathsf{s}}{\sim} \cap \mathcal{U}^-_{S_1,S_2}$.*

Proof: For Part 1 we reason by induction on $(2r - \gamma(S_1, S_2))$. We tackle the inductive step first. Assume the result holds for all (S'_1, S'_2) with $\gamma(S'_1, S'_2) > \gamma(S_1, S_2)$. Let $j' = c(2r - (\gamma(S_1, S_2) + 1) + 1)(|Q|^2 + r^2|Q|) = c(2r - \gamma(S_1, S_2))(|Q|^2 + r^2|Q|)$. Then, for all such (S'_1, S'_2), $(\overset{j'}{\sim} \cap \mathcal{U}^-_{S'_1,S'_2}) = (\overset{\mathsf{s}}{\sim} \cap \mathcal{U}^-_{S'_1,S'_2})$.

Observe that, for $k > j'$, if $\overset{k}{\sim} \cap \mathcal{U}^-_{S_1,S_2} = \overset{k+1}{\sim} \cap \mathcal{U}^-_{S_1,S_2}$, then we must have $\overset{k}{\sim} \cap \mathcal{U}^-_{S_1,S_2} = \overset{\mathsf{s}}{\sim} \cap \mathcal{U}^-_{S_1,S_2}$, because the (SYS) conditions for (S_1, S_2) refer to either (S_1, S_2) or (S'_1, S'_2) with $\gamma(S'_1, S'_2) > \gamma(S_1, S_2)$.Consequently, if $\overset{j'}{\sim} \cap \mathcal{U}^-_{S_1,S_2} \neq \overset{\mathsf{s}}{\sim} \cap \mathcal{U}^-_{S_1,S_2}$, the sequence $(\overset{k}{\sim} \cap \mathcal{U}^-_{S_1,S_2})$ $(k = j', j' + 1, \cdots)$ will have to change in every step before stabilisation. Thus, the steps before stabilisation will induce a subchain of the chain analysed in Lemma 18 (2). Hence, at most $c(|Q|^2 + r^2|Q|)$ extra steps from $(\overset{j'}{\sim})$ will be required to arrive at $\overset{\mathsf{s}}{\sim} \cap \mathcal{U}^-_{S_1,S_2}$, which delivers the required bound.

The base case ($\gamma(S_1, S_2) = 2r$) can be established in a similar fashion: in this case the (SYS) conditions can only refer to (S_1, S_2), thus the sequence $(\overset{k}{\sim} \cap \mathcal{U}^-_{S_1,S_2})$ $(k \geq 0)$ will be strictly decreasing before stabilisation and the bound from Lemma 18 (2) can be applied.

Part 2 follows from Part 1, because $c(2r+1)(|Q|^2+r^2|Q|)$ is the largest of all the bounds. ∎

Proposition 20. *For any RA($S\#_0$) bisimulation problem, if there is a winning strategy for Attacker then there is one of depth $O(r|Q|^2 + r^3|Q|)$.*

Proof: We first observe that bisimulation strategies and their corresponding symbolic bisimulation strategies have the same depth. Thus, it suffices to bound symbolic strategies for Attacker. The $O(r|Q|^2 + r^3|Q|)$ bound follows from Part 2 of the preceding Lemma. ∎

Proposition 21. *$\sim$-RA($S\#_0$) is solvable in PSPACE.*

Proof: In Remark 9 we established that bisimilarity for RA($M\#$) can be reduced to the finite-alphabet case at the cost of prolonging the bisimulation game by a constant factor. Consequently, the polynomial bound from the preceding Proposition (for RA($S\#_0$)) is also valid after the reduction to the finite-alphabet case.

Thanks to the bound, it suffices to play the corresponding bisimulation games for polynomially many steps. The existence of a winning strategy can then be established by an alternating Turing machine running in polynomial time. The PSPACE bounds then follows from APTIME = PSPACE. ∎

Proposition 22. *$\sim$-RA($S\#_0$) is PSPACE-hard.*

Proof (sketch): We reduce from the well-known PSPACE-complete problem of checking validity of totally quantified boolean formulas in prenex conjunctive normal form. Universal quantification and selection of conjuncts is performed by Attacker. For existential quantification and disjunctions, we rely on Defender Forcing. The choices of truth values by both players are recorded in registers by using, for each variable x_i, registers $2i, 2i + 1$, both initialised to $\#$. If a player chooses *true* for x_i, we fill register $2i$ leaving $2i + 1$ empty; we do the opposite otherwise. This makes it possible to arrange for bisimilarity/non-bisimilarity (as appropriate) in the final stage of the game, depending on whether the resulting literal is negated. ∎

V. Language equivalence for $RA(S\#_0)$

The results of the previous section can be used to close an existing complexity gap for deterministic language equivalence of register automata. Recall that, in the non-deterministic case, language equivalence (even universality) is undecidable [19]. In the deterministic case, however, the problem can be solved in PSPACE. Sakamoto [21] conjectured that the language inequivalence problem is not in NP. Below we refute the conjecture, showing that, for RA($S\#_0$), the complexity of deterministic language inequivalence actually matches that of nonemptiness [22].

We call an r-RA($S\#_0$) $\mathcal{A}$ *deterministic* if, for all states q of $\mathcal{A}$: (i) for all $(t, i) \in \Sigma \times [1, r]$ there is at most one transition of the form $q \xrightarrow{t,i} q'$, and (ii) for all $t \in \Sigma$ there is at most one transition of the form $q \xrightarrow{t,i^\bullet} q'$. On the other hand, an LTS is deterministic if, for all $\kappa \in \mathbb{C}$ and $\ell \in \mathcal{A}ct$, there is at most one transition $\kappa \xrightarrow{\ell} \kappa'$. Note that if $\mathcal{A}$ is deterministic then so is its transition system $\mathcal{S}(\mathcal{A})$.[4] Then, from Proposition 20, one obtains the following.

Lemma 23. *Let $\mathcal{A}_i = \langle Q_i, q_{0i}, \rho_{0i}, \delta_i, F_i \rangle$ be a deterministic r_i-RA($S\#_0$) ($i = 1, 2$), $r = \max(r_1, r_2)$ and $N = |Q_1| + |Q_2|$.*

[4]The converse may fail due to transitions of $\mathcal{A}$ not being fireable in $\mathcal{S}(\mathcal{A})$.

If $\mathcal{L}(\mathcal{A}_1) \neq \mathcal{L}(\mathcal{A}_2)$ then there is some $w \in (\mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)) \setminus (\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2))$ with $|w| \in O(rN^2 + r^3N)$.

Theorem 24. *Language inequivalence for deterministic RA($S\#_0$) is NP-complete.*

Proof: Membership in NP is achieved via Lemma 23. NP-hardness follows from NP-completeness of language non-emptiness for deterministic RA($S\#_0$) [22]. ∎

VI. NP BOUND FOR SINGLE ASSIGNMENT WITH FILLED REGISTERS (RA(SF))

In Section IV we showed, in the setting with single assignment and no erasures (denoted by RA($S\#_0$)) the bisimilarity problem was solvable in polynomial space. Here we show that a further improvement is possible in the RA(SF) case, i.e. if the registers are required to be filled from the very start. We shall show an NP upper bound.

We start off with a series of results aiming to identify succinct (polynomial-size) sets of generators for $\overset{\mathsf{s}}{\sim}$, which we shall call *generating systems*. In Section IV we already found that parts of $\overset{\mathsf{s}}{\sim}$ exhibit group-theoretic structure. Namely, Lemma 17 shows that, for any $p \in Q$ and $S \subseteq [1,r]$, $\mathcal{G}^p_S(\overset{\mathsf{s}}{\sim}) = \{\sigma \cap (X^p_S \times X^p_S) \,|\, (p,S) \overset{\mathsf{s}}{\sim}_\sigma (p,S)\}$ is a group, where $X^p_S(\overset{\mathsf{s}}{\sim}) \subseteq S$ is the characteristic set of (p,S).

Note that, for RA(SF), we only have the case $S = [1,r]$. Furthermore, $\overset{\mathsf{s}}{\sim}$ will be the only closed relation that we shall consider. For these reasons, we write simply X^p for characteristic set $X^p_{[1,r]}(\overset{\mathsf{s}}{\sim})$ and $\mathcal{G}^p$ for group $\mathcal{G}^p_{[1,r]}(\overset{\mathsf{s}}{\sim})$.

The group-theoretic structure implies that $\mathcal{G}^p$ can be generated by linearly many generators with respect to r.

Lemma 25 ([16])**.** *Every subgroup of $\mathcal{S}_n$ has at most $\max(2, \lfloor \frac{n}{2} \rfloor)$ generators.*

To handle the more general case $(p,S) \overset{\mathsf{s}}{\sim}_\sigma (q,S)$ of different states, consider

$$\mathcal{K}^{p,q} = \{\sigma \cap (X^p \times X^q) \mid (p,[1,r]) \overset{\mathsf{s}}{\sim}_\sigma (q,[1,r])\}.$$

Observe that, for $\sigma_1, \sigma_2 \in \mathcal{K}^{p,q}$, we have $\sigma_2 = (\sigma_2; \sigma_1^{-1}); \sigma_1$, because $\sigma_1^{-1}; \sigma_1 = \mathsf{id}_{X^q}$. Moreover, $\sigma_2; \sigma_1^{-1} \in \mathcal{G}^p$, so σ_2 has been obtained from σ_1 and an element of $\mathcal{G}^p$. Consequently, in presence of generators of $\mathcal{G}^p$, one member of $\mathcal{K}^{p,q}$ suffices to generate the whole of $\mathcal{K}^{p,q}$ by composition. This observation motivates the following definition of a generating system.

Definition 26. A ***generating system*** $\mathcal{G}$ consists of:

- a partitioning of Q into $P_1, \cdots, P_k$;
- for each partition P_i, a single representative $p_i \in P_i$ and:
  - a characteristic set $X^{p_i} \subseteq [1,r]$;
  - a set G^{p_i}, of up to $\max(2, \lfloor \frac{r}{2} \rfloor)$ permutations $\sigma \in \mathcal{S}_{X^{p_i}}$;
  - for each $q \in P_i \setminus \{p_i\}$, a partial permutation $\mathsf{ray}^{p_i}_q \in \mathcal{IS}_{[1,r]}$ such that $\mathsf{dom}(\mathsf{ray}^{p_i}_q) = X^{p_i}$; for technical convenience, we also add $\mathsf{ray}^{p_i}_{p_i} = \mathsf{id}_{X^{p_i}}$.

We write $rep(\mathcal{G})$ for the set $\{p_1, \cdots, p_k\}$ of representatives.

A generating system is used to generate a relation $gen(\mathcal{G}) \subseteq (Q \times \{[1,r]\} \times \mathcal{IS}_r \times Q \times \{[1,r]\})$ as follows. First, set

$$\begin{aligned}\mathrm{BASE}_\mathcal{G} = &\{(p_i,[1,r],\sigma,p_i,[1,r]) \,|\, p_i \in rep(\mathcal{G}), \sigma \in G^{p_i}\} \\ &\cup \{(p_i,[1,r],\mathsf{ray}^{p_i}_q,q,[1,r]) \,|\, p_i \in rep(\mathcal{G}), q \in P_i\}\end{aligned}$$

and then take $gen(\mathcal{G}) = Cl(\mathrm{BASE}_\mathcal{G})$.

Lemma 27. *There exists a generating system $\mathcal{G}$ such that $gen(\mathcal{G}) = \overset{\mathsf{s}}{\sim}$.*

Proof: We partition Q into equivalence classes defined by: $p \sim q$ if and only if there exists σ such that $(p,[1,r],\sigma,q,[1,r]) \in \overset{\mathsf{s}}{\sim}$. For each equivalence class P_i, we pick a single member p_i arbitrarily and let G^{p_i} consist of the generators of $\mathcal{G}^{p_i}$ provided by Lemma 25. Consider $q \in P_i \setminus \{p_i\}$. Because $q \in P_i$, there exists σ such that $(p_i,[1,r],\sigma,q,[1,r]) \in \overset{\mathsf{s}}{\sim}$. Then we can take $\mathsf{ray}^{p_i}_q = \sigma \cap (X^{p_i} \times [1,r])$. By the previous discussion, this delivers the sought generating system. ∎

Lemma 28. *For any generating system $\mathcal{G}$, membership in $gen(\mathcal{G})$ can be determined in polynomial time.*

Proof: To determine whether $(q_1,[1,r],\sigma,q_2,[1,r]) \in gen(\mathcal{G})$, we proceed as follows. If q_1, q_2 belong to different partitions we return NO. Suppose $q_1, q_2 \in P_i$. Recall that $\mathrm{BASE}_\mathcal{G}$ contains $(p_i,[1,r],\mathsf{ray}^{p_i}_{q_j},q_j,[1,r])$ with $\mathsf{dom}(\mathsf{ray}^{p_i}_{q_j}) = X^{p_i}$. Then $(q_1,[1,r],\sigma,q_2,[1,r]) \in gen(\mathcal{G})$ is equivalent to $(p_i,[1,r],\sigma',p_i,[1,r]) \in gen(\mathcal{G})$, where $\sigma' = \mathsf{ray}^{p_i}_{q_1}; \sigma; (\mathsf{ray}^{p_i}_{q_2})^{-1}$. This is in turn equivalent to $\sigma' \cap (X^{p_i} \times X^{p_i})$ being generated from permutations in G^{p_i}. That the latter problem is solvable in polynomial time is a well-known result in computational group theory [10]. ∎

Theorem 29. *$\sim$-RA(SF) is solvable in NP.*

Proof: First we guess a generating system $\mathcal{G}$ and verify whether $gen(\mathcal{G})$ is a bisimulation. By Lemma 27, there exists at least one generating system with this property. Because generating systems involve polynomially many components of polynomial size, they can be guessed in polynomial time. Next, in order to check whether the guessed generating system generates a bisimulation, we need to verify the (SYS) conditions (for $S_1 = S_2 = [1,r]$) for each of the polynomially many elements of $\mathrm{BASE}_\mathcal{G}$. Note that this will involve polynomially many membership tests for $gen(\mathcal{G})$, each of which can be performed in polynomial time by Lemma 28. If the guess leads to a non-bisimulation, we return NO. Otherwise, we use another membership test for $gen(\mathcal{G})$ to check whether the given instance of the bisimilarity problem belongs to $gen(\mathcal{G})$. We return the outcome of that test as the final result. ∎

Remark 30. Note that symbolic bisimulations are based on *partial finite permutations*, which form inverse semigroups. Consequently, inverse semigroup-theoretic structure could seem the most natural kind of structure with which to approach our problems. Unfortunately, inverse semigroups do not admit analogous results.

- There exist inverse subsemigroups of $\mathcal{IS}_n$ that require

$\binom{n}{\frac{n}{2}} \approx 2^n \sqrt{\frac{2}{\pi n}}$ generators, e.g. $\{\mathsf{id}_X | X \subseteq [1,n], |X| = \frac{n}{2}\}$.
- It is possible to show that the membership problem for inverse subsemigroups of $\mathcal{IS}_n$ is PSPACE-complete, sharpening a result of Kozen [15].

Consequently, we were forced to look a bit deeper, and base generating systems on groups.

Remark 31. Note that we do not have a matching lower bound for RA(SF) which raises the intriguing prospect that there may still be scope for improvement in this case.

VII. Fresh-register automata with single assignment without erasure (FRA($S\#_0$))

In this section we examine the problems tackled in Sections IV-VI albeit in the general case of FRAs. We would like to apply the same techniques, aiming to produce the same upper bounds, yet the FRA setting raises significant additional challenges. Our approach for RAs relied on symbolic bisimulations and the group-theoretic structure that emanated from them. While we can express bisimilarity in FRAs symbolically following [27], we shall see that such symbolic bisimulations do not support the group-theoretic representations. The reason is the treatment of the history of the computation, which affects bisimilarity in subtle ways, especially in the initial stages of the bisimulation game. In those stages, global and local freshness can inter-simulate another, under certain conditions, which leads us to extending our symbolic representations beyond the r names that each system can have in its registers.

Simplified notation: We extend the simplified notation for RA($S\#_0$) by including transition labels for global freshness. Recall that, in any transition $q_1 \xrightarrow{t,X,i,Z} q_2$ of an r-FRA($S\#_0$), we have that $Z = \emptyset$, $i \neq 0$ and $X \in \{\circledast, \emptyset, \{i\}\}$. We thus follow a simpler notation for transitions, with $\delta \subseteq Q \times \Sigma \times \{i, i^\bullet, i^\circledast \mid i \in [1,r]\} \times Q$:

(a) we write each transition $q_1 \xrightarrow{t,\{i\},i,\emptyset} q_2$ as $q_1 \xrightarrow{t,i} q_2$;
(b) and each $q_1 \xrightarrow{t,\emptyset,i,\emptyset} q_2$ as $q_1 \xrightarrow{t,i^\bullet} q_2$;
(c) and each $q_1 \xrightarrow{t,\circledast,i,\emptyset} q_2$ as $q_1 \xrightarrow{t,i^\circledast} q_2$.

(a),(b) are as in RA($S\#_0$). In (c), the automaton reads (t,a) if a is *globally fresh*, i.e. it has not appeared in the history so far, and stores it in register i. Formally, $q \xrightarrow{t,i^\circledast} q'$ can induce a transition $(q,\rho,H) \xrightarrow{t,a} (q', \rho[i \mapsto a], H \cup \{a\})$ just if $a \notin H$.[5]

A. Symbolic bisimulation

Recall that, in the case of RAs, we were able to capture bisimilarity symbolically by using pairs of symbolic configurations of the form $((q_1,S_1),(q_2,S_2))$, whereby S_i represented $\mathsf{dom}(\rho_k)$ of the actual configuration (q_k,ρ_k) represented by (q_k,S_k), and a partial bijection $\sigma : S_1 \to S_2$ capturing the matching names of ρ_1 and ρ_2. Moving to FRAs, the first obstacle we face is that actual configurations contain the full history of names and have therefore unbounded size. For bisimulation purposes, though, keeping track of the whole history, or its size, is not necessary. In fact, history only plays a role in globally fresh transitions and one can easily see that the following rule:

- Every globally fresh transition from q_1 must be matched by a globally or a locally fresh transition from q_2.

is sound for simulation of globally fresh transitions.

However, global freshness leads to severe complications in the simulation of locally fresh transitions. For example, assuming configurations $(q_1,\rho_1,H),(q_2,\rho_2,H)$ with $\mathsf{rng}(\rho_1) = H$, we can see that a transition $q_1 \xrightarrow{t,1^\bullet} q_1'$ can be matched by some $q_2 \xrightarrow{t,1^\circledast} q_2'$, as the local names of q_1 coincide with all the names in H. On the other hand, if $H = \{d_1,d_2\}$ and $\rho_i = \{(1,d_i)\}$ (for $i = 1,2$), then a transition $q_1 \xrightarrow{t,1^\bullet} q_1'$ cannot be matched by some $q_2 \xrightarrow{t,1^\circledast} q_2'$ alone; rather, an additional transition $q_2 \xrightarrow{t,1} q_2''$ is needed in order to capture the fact that $q_1 \xrightarrow{t,1^\bullet} q_1'$ can produce d_2. However, if $|H| > 2r$ then there will always be some $d \in H \setminus (\mathsf{rng}(\rho_1) \cup \mathsf{rng}(\rho_2))$ that can be produced by $q_1 \xrightarrow{t,1^\bullet} q_1'$ and, thence, the only way for q_2 to capture it would be by some locally fresh transition.

From our discussion above it follows that, under certain circumstances which include the fact that $|H| \leq 2r$, local freshness can be captured by global freshness and some known-name transitions. To accommodate this feature, we will design symbolic bisimulations with an additional component $h \in [0,2r] \cup \{\infty\}$ that will abstract the size of $|H|$. The value $h = \infty$ would signify that $|H| > 2r$ and therefore local-fresh cannot be matched by global-fresh. On the other hand, $h \leq 2r$ would mean that $|H| = h \leq 2r$ and therefore extra care would need to be taken for fresh transitions. For $h \leq 2r$, we will consider symbolic configurations (q_i,S_i) $(i = 1,2)$ where $S_i \subseteq [1,3r]$ and $h = |S_i|$, related by bijections $\sigma : S_1 \to S_2$.

- The component $S_i \cap [1,r]$ of S_i will still represent the domain of ρ_i.
- The complementary part $S_i \setminus [1,r]$ will represent the remaining names, those that have passed but no longer reside in ρ_i (i.e. $H \setminus \mathsf{rng}(\rho_i)$), in some canonical fashion.

Effectively, the above will allow us to symbolically represent the history of each FRA, up to the size $2r$, in an ordered way. It will also offer us a way to decide the simulation game for locally fresh transitions. Let us say that one system performs a transition $q_1 \xrightarrow{t,i^\bullet} q_1'$:

1. Such a transition can capture any name d that is represented in some $i' \in S_1 \setminus [1,r]$. If $\sigma(i') \in [1,r]$ then the other system has the name in its registers and can (only) capture it by some $q_2 \xrightarrow{t,\sigma(i')} q_2'$.
2. If $\sigma(i') \in S_2 \setminus [1,r]$ then the name is historical and the other system does not currently have it in its registers. It is therefore obliged to simulate by some locally fresh transition $q_2 \xrightarrow{t,j^\bullet} q_2'$.
3. The transition can also capture any name d that is not in H and, in this case, the other system can capture it by any $q_2 \xrightarrow{t,j^\bullet/j^\circledast} q_2'$. Moreover, such a simulation step would

[5] The latter condition above is slightly different but equivalent to that used in [27]. In *loc. cit.*, the names of ρ are not necessarily included in H and hence in this rule one stipulates that $a \notin \mathsf{rng}(\rho) \cup H$.

increase the size of h by one.

We therefore formulate symbolic bisimulation as follows.

Definition 32. Let $\mathcal{A} = \langle Q, q_0, \rho_0, \delta, F\rangle$ be an r-FRA($S\#_0$). We first set:

$$\mathcal{U}_0 = Q \times \mathcal{P}([1,3r]) \times \mathcal{IS}_{3r} \times Q \times \mathcal{P}([1,3r]) \times ([0,2r] \cup \{\infty\})$$

$$\begin{aligned}\mathcal{U} = \{(q_1, S_1, \sigma, q_2, S_2, h) \in \mathcal{U}_0 \mid\ & \sigma \subseteq S_1 \times S_2 \\ & \wedge\, h \leq 2r \implies |\sigma| = |S_1| = |S_2| = h \\ & \wedge\, h = \infty \implies (\sigma \in \mathcal{IS}_r \wedge S_1, S_2 \subseteq [1,r])\}\end{aligned}$$

A *symbolic simulation* on $\mathcal{A}$ is a relation $R \subseteq \mathcal{U}$, with elements $(q_1, S_1, \sigma, q_2, S_2, h) \in R$ written $(q_1, S_1)\, R^h_\sigma\, (q_2, S_2)$, such that all $(q_1, S_1, \sigma, q_2, S_2, h) \in R$ satisfy the following *fresh symbolic simulation conditions* (FSYS):[6]

(a) for all $q_1 \xrightarrow{t,i} q_1'$,
   1. if $\sigma(i) \in [1,r]$ then there is some $q_2 \xrightarrow{t,\sigma(i)} q_2'$ with $(q_1', S_1)\, R^h_\sigma\, (q_2', S_2)$,
   2. if $\sigma(i) = j' \in [r{+}1, 3r]$ then there is some $q_2 \xrightarrow{t,j^\bullet} q_2'$ with $(q_1', S_1)\, R^h_{(j\,j')\circ\sigma}\, (q_2', (j\,j')\cdot S_2)$,
   3. if $i \in S_1 \setminus \mathsf{dom}(\sigma)$ then there is some $q_2 \xrightarrow{t,j^\bullet} q_2'$ with $(q_1', S_1)\, R^h_{\sigma[i\mapsto j]}\, (q_2', S_2[j])$;

(b) for all $q_1 \xrightarrow{t,i^\bullet} q_1'$, $i' \in S_1 \setminus [1,r]$ and $j \in S_2 \setminus \mathsf{rng}(\sigma)$,
   1. if $\sigma(i') \in [1,r]$ then there is some $q_2 \xrightarrow{t,\sigma(i')} q_2'$ with $(q_1', (i\,i')\cdot S_1)\, R^h_{\sigma\circ(i\,i')}\, (q_2', S_2)$,
   2. if $\sigma(i') = j' \in [r{+}1, 3r]$ then there is some $q_2 \xrightarrow{t,j^\bullet} q_2'$ with $(q_1', (i\,i')\cdot S_1)\, R^h_{(j\,j')\circ\sigma\circ(i\,i')}\, (q_2', (j\,j')\cdot S_2)$,
   3. there exists $q_2 \xrightarrow{t,j} q_2'$ with $(q_1', S_1[i])\, R^h_{\sigma[i\mapsto j]}\, (q_2', S_2)$;

(c) for all $q_1 \xrightarrow{t,\ell_i} q_1'$ with $\ell_i \in \{i^\bullet, i^\circledast\}$ there is some $q_2 \xrightarrow{t,\ell_j} q_2'$ with $\ell_j \in \{j^\bullet, j^\circledast\}$ and,
   1. if $h < 2r$ then, taking $i' = \min([r{+}1, 3r] \setminus S_1)$ and $j' = \min([r{+}1, 3r] \setminus S_2)$, we have $(q_1', (i\,i')\cdot S_1[i'])\, R^{h+1}_{(i\,i')\circ\sigma[i'\mapsto j']\circ(j\,j')}\, (q_2', (j\,j')\cdot S_2[j'])$;
   2. if $h = 2r$ then $(q_1', S_1[i] \cap [1,r])\, R^\infty_{\sigma[i\mapsto j]\cap[1,r]^2}\, (q_2', S_2[j] \cap [1,r])$;
   3. if $h = \infty$ then $(q_1', S_1[i])\, R^\infty_{\sigma[i\mapsto j]}\, (q_2', S_2[j])$ and if $\ell_i = i^\bullet$ then $\ell_j = j^\bullet$.

Here (a3) says that every name that is private to q_1 can be simulated by q_2 with a locally fresh transition. (b3) is its dual: this time it is q_2 playing one of its private names. Finally, (c3) captures the cases where a name that is globally or locally fresh both for q_1 and q_2 is played. Define the inverse by:

$$R^{-1} = \{\, (q_2, S_2, \sigma^{-1}, q_1, S_1, h) \mid (q_1, S_1, \sigma, q_2, S_2, h) \in R \,\}$$

and call R a ***symbolic bisimulation*** if both R and R^{-1} are symbolic simulations. We let *s-bisimilarity*, denoted $\overset{\mathsf{s}}{\sim}$, be the union of all symbolic bisimulations.

As before, we define a sequence of ***indexed bisimilarity*** relations $\overset{i}{\sim} \subseteq \mathcal{U}$ inductively as follows. We let $\overset{0}{\sim}$ be the whole of $\mathcal{U}$. Then, for all $i \in \omega$ and $h \in [0,2r] \cup \{\infty\}$, $(q_1, S_1)\ \overset{i+1}{\sim}{}^h_\tau\ (q_2, S_2)$ just if both $(q_1, S_1, \tau, q_2, S_2, h)$ and $(q_2, S_2, \tau^{-1}, q_1, S_1, h)$ satisfy the (FSYS) conditions in $\overset{i}{\sim}$.

Let $\kappa_i = (q_i, \rho_i, H)$ $(i = 1, 2)$ be configurations with common history H and let $n = |H|$. Their symbolic representation will depend on n. We take $\mathsf{symb}(\kappa_1, \kappa_2) \subseteq \mathcal{U}$ to be:

$$\begin{cases} \{(q_1, \mathsf{dom}(\hat\rho_1), \hat\rho_1; \hat\rho_2^{-1}, q_2, \mathsf{dom}(\hat\rho_2), n) \mid \theta(\hat\rho_1, \hat\rho_2)\} & n \leq 2r \\ \{(q_1, \mathsf{dom}(\rho_1), \rho_1; \rho_2^{-1}, q_2, \mathsf{dom}(\rho_2), \infty)\} & n > 2r \end{cases}$$

where $\theta(\hat\rho_1, \hat\rho_2)$ is the condition stipulating that $\hat\rho_i$ range over all $3r$-register assignments of type $S\#_0$ such that $\mathsf{rng}(\hat\rho_i) = H$ and $\hat\rho_i \restriction [1,r] = \rho_i$, for $i = 1, 2$. In particular, $\mathsf{symb}(\kappa_1, \kappa_2)$ is singleton in case $n > 2r$ but not necessarily so if $n \leq 2r$. The following lemma ensures that, with respect to bisimilarity, the specific choice of element from $\mathsf{symb}(\kappa_1, \kappa_2)$ is not important.

Lemma 33. *For all κ_1, κ_2 as above, if $|H| < 2r$ then either $\mathsf{symb}(\kappa_1, \kappa_2) \subseteq \overset{\mathsf{s}}{\sim}$ or $\mathsf{symb}(\kappa_1, \kappa_2) \cap \overset{\mathsf{s}}{\sim} = \emptyset$.*

Definition 34. We say that κ_1 and κ_2 are *s-bisimilar*, written $\kappa_1 \overset{\mathsf{s}}{\sim} \kappa_2$, if $\mathsf{symb}(\kappa_1, \kappa_2) \subseteq \overset{\mathsf{s}}{\sim}$.

Note how the (FSYS) conditions are divided with respect to the value of h: conditions (a2), (b1), (b2), (c1) and (c2) all require $h \leq 2r$; while conditions (a3), (b3) and (c3) are for $h = \infty$. On the other hand, (a1) applies to all h.

Remark 35. The definition of symbolic bisimulation we give here is crucially more fine-grained than the one in [27]. Although in *loc. cit.* the symbolic bisimulation is also given parametrically to the size of the history h (up to the given bound[7]), for $h \leq 2r$ that formulation is simplistic in that it only keeps track of names that reside in registers of the automata,[8] which in turn prohibits us to derive $(q_1, S_1)\, R^h_{\sigma_1;\sigma_2}\, (q_3, S_3)$ from $(q_1, S_1)\, R^h_{\sigma_1}\, (q_2, S_2)$ and $(q_2, S_2)\, R^h_{\sigma_2}\, (q_3, S_3)$ and apply the group-theoretic approach.

Lemma 36. *Let κ_1 and κ_2 be configurations of an r-FRA($S\#_0$), then: $\kappa_1 \sim \kappa_2 \iff \kappa_1 \overset{\mathsf{s}}{\sim} \kappa_2$. Moreover, for all $i \in \omega$, $\overset{i+1}{\sim} \subseteq \overset{i}{\sim}$ and $(\bigcap_{i\in\omega} \overset{i}{\sim}) = \overset{\mathsf{s}}{\sim}$.*

Similarly to symbolic bisimulations for RA($S\#_0$), we have the following closure properties. Given $R \subseteq \mathcal{U}$ we split R into *components*:

$$R = \sum\nolimits_{h\in[0,2r]\cup\{\infty\}} R^h$$

where $R^h = \{(q_1, S_1, \sigma, q_2, S_2) \mid (q_1, S_1, \sigma, q_2, S_2, h) \in R\}$. We now write $Cl(R)$ for the componentwise closure of R with respect to identity, symmetry, transitivity and extension of partial permutations, i.e. $Cl(R) = \sum_{h\in[0,2r]\cup\{\infty\}} Cl(R^h)$.

Proposition 37. *Symbolic bisimilarity and indexed symbolic bisimilarity for FRA($S\#_0$) are closed.*
1) $Cl(\overset{\mathsf{s}}{\sim}) = \overset{\mathsf{s}}{\sim}$; 2) for all $i \in \omega$: $\overset{i}{\sim} = Cl(\overset{i}{\sim})$.

[6] We say that $(q_1, S_1, \sigma, q_2, S_2, h)$ *satisfies the* (FSYS) *conditions in R.*

[7] In fact, the bound used in [27] is smaller $(2r{-}1)$, due to the fact that it examines bisimulation between configurations with common initial names.

[8] that is, in $(q_1, S_1)\, R^h_\sigma\, (q_2, S_2)$ we always have $S_1, S_2 \subseteq [1,r]$.

We therefore observe that the extension of symbolic representations to the size $3r$, and the ensuing history representation up to size $2r$ along with the extended symbolic bisimulation conditions, have paid off in yielding the desired closure properties. The group-theoretic behaviour of a closed relation R differs between different components:

- R^∞ has the same structure as the closed relations R examined in Section IV-B.
- For $h \in [0, 2r]$, the tuples $(q_1, S_1, \sigma, q_2, S_2) \in R^h$ respect the condition $|S_1| = |S_2| = |\sigma| = h$. In particular, σ is a bijection from S_1 to S_2 and, hence, in this case closure under extension is trivial, and so are characteristic sets ($X^p_S(R^h) = S$). Moreover, $\sigma \in \mathcal{IS}_{3r}$ and $S_1, S_2 \subseteq [1, 3r]$.

We can hence see that the same groups arise as in the case of RA($S\#_0$), and actually simpler in the case $h \in [0, 2r]$, albeit parameterised over h. This allows for a similar group-theoretic treatment.

B. PSPACE bound for bisimulation game

Lemma 38. *Let $h \in [0, 2r] \cup \{\infty\}$, $S_1, S_2 \subseteq [1, 3r]$ and $\mathcal{U}^h_{S_1,S_2} = Q \times \{S_1, S_2\} \times \mathcal{IS}_r \times Q \times \{S_1, S_2\} \times \{h\}$. Then the sub-chain $\{\overset{i}{\sim} \mid (\overset{i+1}{\sim}{}^h \cap\, \mathcal{U}^h_{S_1,S_2}) \subsetneq (\overset{i}{\sim}{}^h \cap\, \mathcal{U}^h_{S_1,S_2})\}$ has size $O(|Q|^2 + r^2|Q|)$.*

Given $S_1, S_2 \subseteq [1, 3r]$ and $h \in [0, 2r] \cup \{\infty\}$, let us call the triple (S_1, S_2, h) ***proper*** just if: either $|S_1| = |S_2| = h$, or $h = \infty$ and $S_1, S_2 \subseteq [1, r]$. For such (S_1, S_2, h), let us define:

$$\hat{\gamma}(S_1, S_2, h) = \begin{cases} \gamma(S_1 \cap [1, r], S_2 \cap [1, r]) + h & \text{if } h \in [0, 2r] \\ \gamma(S_1, S_2) + 2r + 1 & \text{if } h = \infty \end{cases}$$

The measure $\hat{\gamma}$ enables us to show the following bound for stabilising indexed bisimulation, proven similarly to Lemma 19.

Lemma 39. *Let $\mathcal{U}^{h-}_{S_1,S_2} = Q \times \{S_1\} \times \mathcal{IS}_r \times Q \times \{S_2\} \times \{h\}$ and let $\hat{c}$ be the constant of $O(|Q|^2 + r^2|Q|)$ in Lemma 38 (2).*

1) *For any proper (S_1, S_2, h), we have $(\overset{j}{\sim} \cap\, \mathcal{U}^{h-}_{S_1,S_2}) = (\overset{\mathsf{s}}{\sim} \cap\, \mathcal{U}^{h-}_{S_1,S_2})$, where $j = \hat{c}(4r - \hat{\gamma}(S_1, S_2, h) + 2)(|Q|^2 + r^2|Q|)$.*
2) *Let $B = \hat{c}(4r + 2)(|Q|^2 + r^2|Q|)$. For any proper (S_1, S_2, h), it holds that $\overset{B}{\sim} \cap\, \mathcal{U}^{h-}_{S_1,S_2} = \overset{\mathsf{s}}{\sim} \cap\, \mathcal{U}^{h-}_{S_1,S_2}$.*

We can therefore establish PSPACE solvability.

Proposition 40. *For any FRA($S\#_0$) bisimulation problem, if there is a winning strategy for Attacker then there is one of depth $O(r|Q|^2 + r^3|Q|)$.*

Proposition 41. *$\sim$-FRA($S\#_0$) is solvable in PSPACE.*

C. Generating systems and NP routines

We proceed to generating systems for FRA(SF), which are h-parameterised versions of the ones for RA(SF), except that now they are built over $[1, 3r]$ rather than $[1, r]$. Since we again consider only characteristic sets and groups with relation parameter $R = \overset{\mathsf{s}}{\sim}$, we will typically leave this argument implicit in what follows. We call a pair (S, h) proper just if (S, S, h) is proper.

Definition 42. A ***generating system*** $\mathcal{G}_{S,h}$ for proper (S, h) (in which case $|S| \leq 2r$), consists of:

- a partitioning of Q into $P_1, \cdots, P_k$;
- for each partition P_i, a single representative $p_i \in P_i$ and:
  - a characteristic set $X^{p_i}_{S,h} \subseteq S$;
  - a set $G^{p_i}_{S,h}$, of up to $\max(2,r)$ permutations $\sigma \in \mathcal{S}_{X^{p_i}_{S,h}}$;
  - for each $q \in P_i \setminus \{p_i\}$, a partial permutation $\mathsf{ray}^{p_i}_q \in \mathcal{IS}_S$ such that $\mathsf{dom}(\mathsf{ray}^{p_i}_q) = X^{p_i}_{S,h}$; for technical convenience, we also add $\mathsf{ray}^{p_i}_{p_i} = \mathsf{id}_{X^{p_i}_{S,h}}$.

We write $rep(\mathcal{G}_{S,h})$ for the set $\{p_1, \cdots, p_k\}$ of representatives. From $\mathcal{G}_{S,h}$ we generate $gen(\mathcal{G}_{S,h}) \subseteq (Q \times \{S\} \times \mathcal{IS}_{3r} \times Q \times \{S\})$ by setting

$$\begin{aligned} \mathrm{BASE}_{\mathcal{G}_{S,h}} = &\{(p_i, S, \sigma, p_i, S) \mid p_i \in rep(\mathcal{G}_{S,h}) \wedge \sigma \in G^{p_i}_{S,h}\} \\ &\cup \{(p_i, S, \mathsf{ray}^{p_i}_q, q, S) \mid p_i \in rep(\mathcal{G}_{S,h}) \wedge q \in P_i\} \end{aligned}$$

and taking $gen(\mathcal{G}_{S,h}) = \mathit{Cl}(\mathrm{BASE}_{\mathcal{G}_{S,h}})$.

The following lemma, proved in the same way as Lemmata 27 and 28, enables us to prove an NP upper bound for bisimilarity in FRA(SF).

Lemma 43. *1) For any proper (S, h) there exists a generating system $\mathcal{G}_{S,h}$ such that $gen(\mathcal{G}_{S,h}) = \overset{\mathsf{s}}{\sim} \cap\, \mathcal{U}^h_{S,S}$.*
2) For any generating system $\mathcal{G}_{S,h}$, membership in $gen(\mathcal{G}_{S,h})$ can be determined in polynomial time.

Theorem 44. *$\sim$-FRA(SF) is solvable in NP.*

Proof: Given an input tuple $(q_1, S_1, \sigma, q_2, S_2, h^0)$, note first that $[1, r] \subseteq S_1, S_2$ (by F) and $|S_1| = |S_2|$. We can therefore convert it to an equivalent $(q_1, S'_1, \sigma', q_2, S_2, h^0)$, with $S'_1 = S_2$, by applying a permutation on the indices in $S_1 \setminus [1, r]$. Hence, we can assume wlog that our input is some $(q_1, S^0, \sigma, q_2, S^0, h^0)$. Moreover, because the expansion of S in the symbolic bisimulation game (when $h \in [0, 2r]$) always occurs in its first free register ($\min([r+1, 3r] \setminus S)$), we can compute the sequence $(S^0, h^0, S^0), (S^1, h^0+1, S^1), \cdots$ of distinct triples considered in the game (in the $h \in [0, 2r]$ phase), which must thence be bounded in length by $2r$. Including the final bisimulation phase ($h = \infty$), this gives us $2r + 1$ phases. We first generate for each of them a generating system, say $\mathcal{G}_{S^i,h^i}$, and then verify whether each $gen(\mathcal{G}_{S^i,h^i})$ is a symbolic bisimulation, similarly to Theorem 29. Note that each such check can be achieved in polynomial time. If the guess leads to some $gen(\mathcal{G}_{S^i,h^i})$ being a non-symbolic-bisimulation, we return NO. Otherwise, we use another membership test for $gen(\mathcal{G}_{S^0,h^0})$ to check whether the given instance of the bisimilarity problem belongs to $gen(\mathcal{G}_{S^0,h^0})$. We return the outcome of that test as the final result. ■

VIII. Visibly pushdown automata with single assignment and filled registers (VPDRA(SF))

Finally, we consider a variant of register automata with visible pushdown storage [2]. We only consider the most restrictive register discipline (SF), as undecidability will be shown to apply already in this case.

Definition 45. A ***visibly pushdown r-register automaton*** (r-VPDRA(SF)) $\mathcal{A}$ is a tuple $\langle Q, \Sigma_C, \Sigma_N, \Sigma_R, \Gamma, \rho_I, \delta\rangle$, where Q, ρ_I have the same meaning as for r-RA,

- Σ_C, Σ_N, Σ_R are disjoint finite sets of *push-*, *no-op-* and *pop-tags* respectively;
- Γ is a finite set of *stack tags*;
- $\delta = \delta_C \cup \delta_N \cup \delta_R$, the transitions, have $Lab = \{1, \ldots, r\} \cup \{1^\bullet, \ldots, r^\bullet\}$ and:
  - $\delta_C \subseteq Q \times \Sigma_C \times Lab \times \Gamma \times \{1, \cdots, r\} \times Q$
  - $\delta_N \subseteq Q \times \Sigma_N \times Lab \times Q$
  - $\delta_R \subseteq Q \times \Sigma_R \times Lab \times \Gamma \times \{1, \cdots, r, \bullet\} \times Q$

Configurations of r-VPDRA(SF) are triples (q, ρ, s), where $q \in Q$, ρ is a register assignment and $s \in (\Gamma \times \mathcal{D})^*$ is the stack. An LTS arises by having a labelled edge $(q_1, \rho_1, s_1) \xrightarrow{(t,d)} (q_2, \rho_2, s_2)$ just if there exist $i \in [1, r]$ and $l \in \{i, i^\bullet\}$ such that: (i) $\rho_1(x) = \rho_2(x)$ for all $x \neq i$; (ii) if $l = i$ then $\rho_1(i) = \rho_2(i)$, otherwise $\rho_2(i) \notin \mathsf{rng}(\rho_1)$; and (iii) one of the following conditions holds:

- $(q_1, t, l, t', j, q_2) \in \delta_C$ and $s_2 = (t', \rho_2(j))s_1$,
- $(q_1, t, l, q_2) \in \delta_N$ and $s_2 = s_1$,
- $(q_1, t, l, t', j, q_2) \in \delta_R$, $s_1 = (t', d')s_2$,

where if $j \in [1, r]$ then $d' = \rho_2(j)$, otherwise $d' \notin \mathsf{rng}(\rho_2)$.

We show that even the visibly pushdown with SF register discipline is undecidable. To do so, we reduce from the undecidable emptiness problem for (one-way) *universal* register automata with two registers (URA_2) [9].

Theorem 46. *VPDRA(SF) bisimiliarity is undecidable.*

Proof (sketch): Given a URA_2 U, we devise a 2-VPDRA $\mathcal{A}_U$ with two configurations κ_1, κ_2 such that U accepts a word iff $\kappa_1 \not\sim \kappa_2$. $\mathcal{A}_U$ is constructed to induce a bisimulation game in which Attacker gets a chance to choose a word to be accepted by U and simulate an accepting run (if one exists). The stack of $\mathcal{A}_U$ is used to store the word that Attacker has chosen, with the top of the stack playing the role of the head of U and the two registers of $\mathcal{A}_U$ emulating the two registers of U. To simulate a transition we arrange for Attacker to guess the outcome of the comparison of the top of stack with the current register contents whilst allowing Defender to verify the correctness of such guesses via Defender forcing. Transitions from universal states are chosen by Defender, again using Defender forcing. ∎

The argument sketched above also reduces URA_1 emptiness to 1-VPDRA, which implies a non-primitive-recursive lower bound for 1-VPDRA.

IX. Conclusion

We have demonstrated bounds on the bisimilarity problem for broad classes of (fresh-)register automata, which include those studied in the literature. The ability to start with empty registers, erase their contents (or equivalently, store duplicate values) and use of a stack all affect the inherent problem complexity. Global freshness, however, does not seem to affect complexity. Except for the SF discipline, all bounds are tight.

Although our problem formulation is with respect to two configurations of a single automaton, extending our results to problems concerning two automata is unproblematic. If the automata have different numbers of registers, the game can be played on an automaton with a number equal to the larger of the two, with additional registers initialised (and left) empty. Even in F register disciplines our arguments show that, since these extra registers are never assigned to, the system can be treated as a $\#_0$ system without change in complexity.

Acknowledgement

We would like to thank M. Jerrum, R. Gray, J. Mitchell and M. Beaudry for useful discussions regarding computational group theory. Thanks also to the anonymous referees for many helpful suggestions. Research supported by the Engineering and Physical Sciences Research Council (EP/J019577/1) and the Royal Academy of Engineering (RF: Tzevelekos).

References

[1] R. Alur, P. Cerný, and S. Weinstein. Algorithmic analysis of array-accessing programs. In *CSL*, *LNCS*, pp. 86–101. Springer, 2009.
[2] R. Alur and P. Madhusudan. Visibly pushdown languages. In *STOC*, pp. 202–211, 2004.
[3] M. F. Atig, A. Bouajjani, and S. Qadeer. Context-bounded analysis for concurrent programs with dynamic creation of threads. *Log. Meth. in Comput. Sci.*, 7(4), 2011.
[4] L. Babai. On the length of subgroup chains in the symmetric group. *Com. in Algebra*, 14(9), 1986.
[5] M. Benedikt, S. Göller, S. Kiefer, and A. S. Murawski. Bisimilarity of pushdown automata is nonelementary. In *LICS*, pp. 488–498. IEEE Computer Society, 2013.
[6] M. Bojanczyk, B. Klin, and S. Lasota. Automata theory in nominal sets. *Log. Meth. in Comp. Sci.*, 10(3), 2014.
[7] M. Boreale and L. Trevisan. A complexity analysis of bisimilarity for value-passing processes. *Theor. Comput. Sci.*, 238(1-2):313–345, 2000.
[8] V. Ciancia and U. Montanari. Symmetries, local names and dynamic (de)-allocation of names. *Inf. and Comp.*, 208(12):1349 – 1367, 2010.
[9] S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Log.*, 10(3), 2009.
[10] M. L. Furst, J. E. Hopcroft, and E. M. Luks. Polynomial-time algorithms for permutation groups. In *FOCS*, pp. 36–41. IEEE Comp. Soc., 1980.
[11] R. Grigore, D. Distefano, R. L. Petersen, and N. Tzevelekos. Runtime verification based on register automata. In *TACAS*, *LNCS*, pp. 260–276. Springer, 2013.
[12] P. Jančar and J. Srba. Undecidability of bisimilarity by defender's forcing. *J.ACM*, 55(1), 2008.
[13] B. Jonsson and J. Parrow. Deciding bisimulation equivalences for a class of non-finite-state programs. *Inf. Comput.*, 107(2):272–302, 1993.
[14] M. Kaminski and N. Francez. Finite-memory automata. *Theor. Comput. Sci.*, 134(2), 1994.
[15] D. Kozen. Lower bounds for natural proof systems. In *FOCS*, pp. 254–266. IEEE, 1977.
[16] A. McIver and P. M. Neumann. Enumerating finite groups. *Quart. J. Math.*, 38(4), 1987.
[17] U. Montanari and M. Pistore. An introduction to history dependent automata. *Electr. Notes Theor. Comput. Sci.*, 10, 1997.
[18] A. S. Murawski, S. J. Ramsay, and N. Tzevelekos. Reachability in pushdown register automata. In *MFCS*, *LNCS*, pp. 464–473. Springer, 2014.
[19] F. Neven, T. Schwentick, and V. Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Log.*, 5(3):403–435, 2004.
[20] M. Pistore. *History Dependent Automata*. PhD thesis, University of Pisa, 1999.
[21] H. Sakamoto. *Studies on the Learnability of Formal Languages via Queries*. PhD thesis, Kyushu University, 1998.
[22] H. Sakamoto and D. Ikeda. Intractability of decision problems for finite-memory automata. *Theor. Comput. Sci.*, 231(2):297–308, 2000.
[23] L. Segoufin. Automata and logics for words and trees over an infinite alphabet. In *CSL*, *LNCS*, pp. 41–57. Springer, 2006.
[24] G. Sénizergues. The bisimulation problem for equational graphs of finite out-degree. *SIAM J. Comput.*, 34(5):1025–1106, 2005.
[25] J. Srba. Visibly pushdown automata: From language equivalence to simulation and bisimulation. In *CSL*, *LNCS*, pp. 89–103. Springer, 2006.
[26] J. Srba. Roadmap of infinite results. `http://www.brics.dk/~srba/roadmap/`, 2008.
[27] N. Tzevelekos. Fresh-register automata. In *POPL*, pp. 295–306. ACM Press, 2011.