Revisiting Membership Problems in Subclasses of Rational Relations

Pascal Bergsträßer
Department of Computer Science
RPTU Kaiserslautern-Landau
Kaiserslautern, Germany

Moses Ganardi
Max Planck Institute for Software Systems
(MPI-SWS)
Kaiserslautern, Germany

Abstract—We revisit the *membership problem* for subclasses of rational relations over finite and infinite words: Given a relation R in a class $\mathbf{C}_2$, does R belong to a smaller class $\mathbf{C}_1$? The subclasses of rational relations that we consider are formed by the deterministic rational relations, synchronous (also called automatic or regular) relations, and recognizable relations. For almost all versions of the membership problem, determining the precise complexity or even decidability has remained an open problem for almost two decades. In this paper, we provide improved complexity and new decidability results. (i) Testing whether a synchronous relation over infinite words is recognizable is NL-complete (PSPACE-complete) if the relation is given by a deterministic (nondeterministic) ω-automaton. This fully settles the complexity of this recognizability problem, matching the complexity of the same problem over finite words. (ii) Testing whether a deterministic rational binary relation is recognizable is decidable in polynomial time, which improves a previously known double exponential time upper bound. For relations of higher arity, we present a randomized exponential time algorithm. (iii) We provide the first algorithm to decide whether a deterministic rational relation is synchronous. For binary relations the algorithm even runs in polynomial time.

I. INTRODUCTION

The study of *relations over words* and their computational models, often called *transducers*, has become an active field of research, with applications in various fields, including algorithmic verification [1], [2], synthesis [3], [4], and graph databases [5]. While the class of regular languages is captured by several equivalent automata models, e.g. deterministic and nondeterministic automata, which read their input either in one or both directions, the same does not hold anymore for relations. The literature contains a number of transducer models for relations with varying tradeoffs between expressivity, closure properties, and algorithmic amenability. Finite-state transducers reading multiple words have been already introduced by Rabin and Scott in their seminal paper [6]. This basic model has later been extended to more expressive models such as streaming string transducers, transductions with origin semantics, visibly pushdown transducers, and transducers over infinite words, see [7] for an overview. Various algorithmic questions on basic transducer models remain challenging open problems, e.g. determining the precise complexity of the equivalence problem for deterministic streaming string transducers [1] or for deterministic multitape automata [8].

The membership problem: In this paper we revisit the *membership problem* (or the *definability problem*) for relations over words, i.e. given a relation R in a class $\mathbf{C}_2$, does R belong to a smaller class $\mathbf{C}_1$? The membership problem for languages is a classical question in automata theory, in particular the question whether a given regular language belongs to a subclass of the regular languages [9], [10], [11], [12], [13], and the question whether a given language from a superclass of the regular languages is in fact regular [14], [15], [16]. For example, Schützenberger's theorem effectively characterizes which regular languages are star-free [9]. Deciding whether an NFA accepts a star-free language is PSPACE-complete [17]. Another milestone result in this context is Valiant's regularity test for deterministic pushdown automata (DPDAs) [15]. Its running time is double exponential, improving on a previous triple exponential time algorithm by Stearns [18]. The only known lower bound is P-hardness inherited from emptiness problem, leaving an almost fifty year old double exponential gap between the upper and the lower bound.

The membership problem for relations over words was first systematically studied by Carton, Choffrut, and Grigorieff [19] for subclasses of rational relations over finite words. Let us briefly introduce the most important subclasses, see Section II for formal definitions. A relation is *rational* if it is recognized by a nondeterministic multitape automaton where the tapes are read asynchronously in one direction [20]. The deterministic variant of multitape automata [6] captures the class of *deterministic rational* relations. Unfortunately, universality of rational relations and inclusion of deterministic rational relations are undecidable [21]. To overcome these undecidability barriers, one can put the additional restriction on the automaton that all heads read their input letter synchronously in parallel. Synchronous multitape automata recognize the *synchronous (rational) relations* [22], also called automatic or regular relations. Due to their effective closure under first-order operations, they enjoy pleasant algorithmic properties and form the basis of *automatic structures* [23], [24] and of *regular model checking* [25], [26]. The smallest class of relations we consider is formed by the *recognizable relations*, where the input words are processed by independent automata that synchronize only on the sequence of final states reached after reading the entire words. Alternatively, recognizable relations can be described as finite unions of Cartesian products

of regular languages [27, Theorem 1.5]. All mentioned classes of relations over finite words are extended to infinite words, by adding a Büchi condition for nondeterministic automata or a parity condition for deterministic automata. The hierarchy of the considered subclasses of rational relations over finite and infinite words is displayed in Figure 1.

However, as most interesting problems on rational relations, it is undecidable to test whether a given rational relation is recognizable, synchronous, or deterministic rational [21], [28]. Hence, we turn our attention to subclasses of deterministic rational relations. The following observation from [19] makes a simple connection between *binary* rational relations $R \subseteq \Sigma^* \times \Sigma^*$ and context-free languages: If R is rational then $L_R = \{\mathsf{rev}(u)\#v \mid (u,v) \in R\}$ is context-free where $\mathsf{rev}(u)$ is the reversal of u; if R is deterministic rational then L_R is deterministic context-free. Furthermore, R is recognizable if and only if L_R is regular. Therefore, recognizability of binary deterministic rational relations can be easily reduced to regularity of DPDAs, which can be decided in double exponential time [15]. Using methods from the regularity algorithm, originally due to Stearns [18], one can also decide[1] recognizability of deterministic rational relations of arbitrary arity [19]. Carton, Choffrut, and Grigorieff also present an algorithm to test whether a synchronous relation is recognizable [19], which runs in double exponential time (see the remark on its running time in [29]). Recently, Barceló et al. determined the precise complexity of the same problem [30], see below. The question how to decide whether a deterministic rational relation is synchronous still hitherto remains open.

Recognizability and the infinite clique problem: It has been observed in [19] that the recognizability problem for subclasses of rational relations can be reduced to checking whether certain equivalence relations have finite index. For a relation $R \subseteq (\Sigma^*)^k$ and $j \in [1, k-1]$ define the equivalence relations $\sim_j^R$ on $(\Sigma^*)^j$ by

$$\boldsymbol{x} \sim_j^R \boldsymbol{y} \stackrel{\text{def}}{\iff} \text{for all } \boldsymbol{z} \in (\Sigma^*)^{k-j} :$$
$$[(\boldsymbol{x}, \boldsymbol{z}) \in R \iff (\boldsymbol{y}, \boldsymbol{z}) \in R],$$

resembling the Myhill-Nerode congruence for languages. If R is rational, then R is recognizable if and only if $\sim_j^R$ has finite index for all $j \in [1, k-1]$ [19, Proposition 3.8]. This characterization has been used in [30] to decide recognizability for synchronous relations, as follows. Given a DFA (NFA) for a synchronous relation R, one can compute automata for the complement relations $\not\sim_j^R$ in logarithmic space (polynomial space). Hence, to decide non-recognizability of R it suffices to test whether for a given *co-equivalence relation* $\not\sim$ there exists an infinite set X such that $\boldsymbol{x} \not\sim \boldsymbol{y}$ for all distinct $\boldsymbol{x}, \boldsymbol{y} \in X$, in other words, whether $\not\sim$ has an infinite clique. In fact, the infinite clique problem for arbitrary synchronous relations was shown to be NL-complete [30] (later simplified in [31]).

However, in certain settings we need to exploit the fact that $\not\sim_j^R$ is the complement of an equivalence relation $\sim_j^R$.

[1]While the authors of [19] did not analyze the complexity of their algorithm, it is easy to see that their algorithm runs in elementary time for fixed arity k.

For example, Löding and Spinrath [29] have shown that the infinite clique problem for ω-synchronous co-equivalence relations is decidable in double exponential time. This yields a double (triple) exponential time algorithm for the ω-recognizability problem for ω-synchronous relations given by (non)deterministic ω-automata. Whether the infinite clique problem over *arbitrary* ω-synchronous relations is decidable is a longstanding open problem [32]. Another example where the difference between co-equivalence relations and arbitrary relations becomes apparent is the case of *tree-automatic relations*. It was proven in [31] that the infinite clique problem for tree-automatic relations is EXP-complete; however, restricted to complements of transitive relations the infinite clique problem becomes P-complete. This yields optimal complexity for the recognizability problem for tree-automatic relations: Recognizability is P-complete for relations given by deterministic bottom-up or top-down tree automata, and EXP-complete for nondeterministic tree automata.

Contributions: We provide improved complexity and new decidability results for the membership problems in subclasses of rational relations over finite and infinite words. To do so, we refine the existing analyses in [19], [29] and identify patterns in the transducers which witness *non*-membership in the subclass. As our first main result, we pinpoint the precise complexity of the ω-recognizability problem of ω-synchronous relations.

Theorem 1. *Given an ω-synchronous relation R by a deterministic parity (resp. nondeterministic Büchi) automaton, it is* NL-*complete (resp.* PSPACE-*complete) to decide whether R is ω-recognizable.*

This matches the complexity of the recognizability problem of synchronous relations over finite words [30]. To prove Theorem 1, we follow the approach of [29] and solve the infinite clique problem for ω-synchronous co-equivalence relations $\not\sim$. Their algorithm constructs an automaton for a regular set of (ultimately periodic) representatives of $\sim$, whose size is double (triple) exponential in the size of a (non)deterministic automaton for $\not\sim$. We circumvent the construction of this large automaton and identify a simple pattern directly in the automaton for $\not\sim$ which witnesses an infinite clique.

Our second and third main result concerns decision problems on deterministic rational relations over finite words. We encounter two issues when applying the same reduction to the infinite clique problem on $\not\sim_j^R$. If R is a binary relation, then it is not difficult to see that $\not\sim_1^R$ is effectively rational since two runs on pairs (x, z) and (y, z) with a common second component z can be simulated in parallel by a 3-tape automaton reading (x, y, z). However, to the best of the authors' knowledge, it is unknown whether the infinite clique problem for rational relations is decidable at all, even when restricted to co-equivalence relations. Moreover, if R has arity $k > 2$ then it is unclear whether the relations $\not\sim_j^R$ are still rational. Instead of reducing to an infinite clique problem, we revisit the proof from [19] and obtain the following improved complexity bounds.
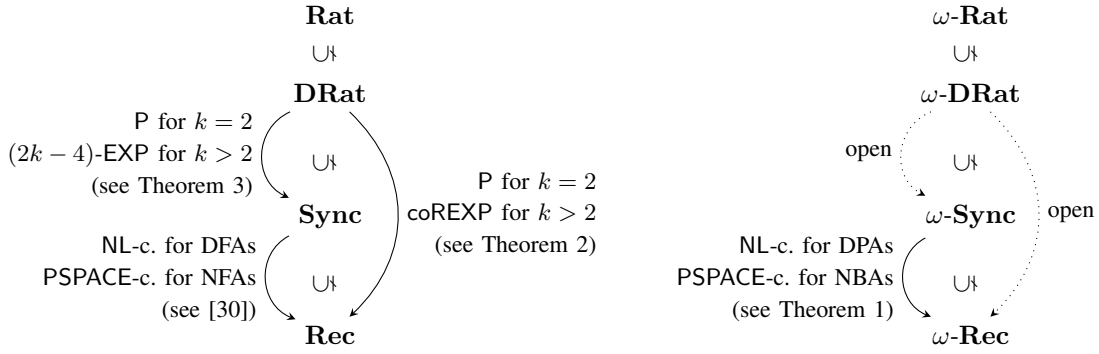
Fig. 1. The complexity landscape of deciding membership to subclasses of rational relations over finite and infinite words. An arrow from $\mathbf{C}_2$ to $\mathbf{C}_1$ refers to the membership problem, given a relation from $\mathbf{C}_2$, does it belong to $\mathbf{C}_1$? Membership of rational relations in any one of the three subclasses is undecidable [21], [28]. Dotted arrows mean that decidability of the problem is unknown.

Theorem 2. *Given a k-ary deterministic rational relation R, one can decide whether R is recognizable (i) in* P *if $k = 2$, (ii) in* coREXP *if $k > 2$ is fixed, and (iii) in* coNEXP *if k is part of the input.*

Here, coREXP $\subseteq$ coNEXP is the class of all decision problems that can be solved by a randomized algorithm in exponential time, which may err on negative instances with probability at most $1/2$. To show Theorem 2 we reduce the recognizability problem to the equivalence problem of deterministic k-tape automata. The reduction works in logspace if $k = 2$, but requires polynomial space for $k > 2$. Let us remark that the precise complexity of the equivalence problem is unknown: Harju and Karhumäki showed that testing equivalence of deterministic rational relations is in coNP [33]. Moreover, Friedman and Greibach devised a polynomial time algorithm for binary relations [34], and for fixed arity $k > 2$ equivalence is decidable in randomized polynomial time [8]. For the reduction, we first observe that recognizability can also be described in terms of modified equivalence relations $\approx_j^R$ over words instead of $\sim_j^R$, which is defined over j-tuples of words. Second, we extract from [19] an automaton pattern that witnesses nonrecognizability. In addition, for the case of binary relations, we need the simple but crucial observation mentioned above that two runs on pairs of words with a common component can be simulated in parallel.

In addition, we observe that over deterministic rational relations the equivalence problem is logspace reducible to the recognizability problem (Theorem 8). Essentially, this follows from a result by Friedman and Greibach [35], which reduces the equivalence problem of DPDAs restricted to a subclass $\mathbf{C}$ to the membership problem of DPDAs to $\mathbf{C}$. Hence, over binary deterministic rational relations the recognizability problem and the equivalence problem are in fact *logspace interreducible*.

Moreover, we present a construction that transforms a deterministic multitape automaton into an equivalent double exponentially sized *independent multitape automaton*, assuming it exists, i.e. if the relation is recognizable. This provides an answer to the problem of how to compute *monadic decom-*

positions for deterministic rational and synchronous relations, see [30, Section 6]. The construction is based on known ideas from [19] and imitates Valiant's construction of a double exponentially large DFA from a regular DPDA [15]. It seems that the missing piece for the construction is our characterization of recognizability via the equivalence relations $\approx_j^R$.

Finally, we prove that one can decide whether a deterministic rational relation is synchronous by a reduction to the recognizability problem, which was left open in [19].

Theorem 3. *Given a k-ary deterministic rational relation R, one can decide whether R is synchronous (i) in* P *if $k = 2$ and (ii) in $(2k-4)$-*EXP *if $k > 2$.*

The intuition behind the algorithm is that the heads of a deterministic multitape automaton for a synchronous relation must have *bounded delay* throughout the computation, see [22, Section 3], except if, from some point on, the components are independent from each other. To check the latter condition, we need the recognizability test from Theorem 2.

Applications: As a corollary of our results on ω-synchronous relations we will provide a PSPACE-algorithm which tests whether a quantifier-free formula over mixed real-integer linear arithmetic $(\mathbb{R}; \mathbb{Z}, +, <, 0, 1)$ is *monadically decomposable*, i.e. equivalent to a Boolean combination of monadic formulas.

Recently, recognizable relations have been featured in a decidable string constraint language, motivated by the verification of string-manipulating programs [2]. One semantic condition of the constraint language requires that the relations appearing in the constraint are *effectively* recognizable, i.e. one can compute a representation as a union of Cartesian products of regular languages. If the given relations are deterministic rational, we can effectively decide recognizability (in polynomial-time for binary relations) and compute the required representation in double exponential time.

II. RATIONAL RELATIONS AND THEIR SUBCLASSES

In the following we introduce the classes of rational relations, deterministic rational relations, synchronous relations, and recognizable relations, which are denoted by $\mathbf{Rec} \subseteq$

Sync $\subseteq$ **DRat** $\subseteq$ **Rat**. Similarly, on infinite words we consider ω-rational relations, deterministic ω-rational relations, ω-synchronous relations, and ω-recognizable relations, denoted by ω-**Rec** $\subseteq$ ω-**Sync** $\subseteq$ ω-**DRat** $\subseteq$ ω-**Rat**. Since ω-**DRat** and ω-**Rat** will not be used in this work, we will not define these classes.

Let Σ be a finite alphabet. The product of k free monoids $(\Sigma^*)^k$ forms a monoid with componentwise multiplication $(u_1, \ldots, u_k)(v_1, \ldots, v_k) = (u_1 v_1, \ldots, u_k v_k)$. We often denote word tuples by boldface letters $\boldsymbol{u}$ and denote its i-th entry by u_i. As usual we identify a pair of tuples $(\boldsymbol{u}, \boldsymbol{v})$ with the concatenation of $\boldsymbol{u}$ and $\boldsymbol{v}$. Furthermore $\boldsymbol{\varepsilon} = (\varepsilon, \ldots, \varepsilon)$ denotes a tuple of empty words of appropriate dimension. The *length* of a word tuple $\|\boldsymbol{u}\| = \sum_{i=1}^{k} |u_i|$ is the total length of its entries. We assume familiarity with the basic models of (non)deterministic finite automata over finite and infinite words. Recall that the class of ω-regular languages is described by *nondeterministic Büchi automata* (NBAs) as well as by *deterministic parity automata* (DPAs) [36, Section 1].

a) Rational relations: A k-tape automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ consists of a finite state set Q, a finite alphabet Σ, an initial state q_0, a set $F \subseteq Q$ of final states, and a finite set of transitions $\Delta \subseteq Q \times (\Sigma^*)^k \times Q$. A run of $\mathcal{A}$ on a tuple $\boldsymbol{w} \in (\Sigma^*)^k$ from p_0 to p_n is a sequence of transitions $p_0 \xrightarrow{\boldsymbol{w}_1} p_1 \xrightarrow{\boldsymbol{w}_2} \cdots \xrightarrow{\boldsymbol{w}_n} p_n$ with $\boldsymbol{w} = \boldsymbol{w}_1 \boldsymbol{w}_2 \cdots \boldsymbol{w}_n$. The relation $R(\mathcal{A})$ accepted by $\mathcal{A}$ consists of all tuples $\boldsymbol{w} \in (\Sigma^*)^k$ such that $\mathcal{A}$ has a run on $\boldsymbol{w}$ from the initial to a final state. Relations accepted by k-tape automata are called *rational*.

b) Deterministic rational relations: For k-tape automata we define the sets $H_1, \ldots, H_k$ by $H_i = \{\varepsilon\}^{i-1} \times \Sigma \times \{\varepsilon\}^{k-i}$. A k-tape automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ is *deterministic* if (i) Q is equipped with a partition into sets $Q = \bigcup_{i=1}^{k} Q_i$, (ii) the transition relation has the form $\Delta \subseteq \bigcup_{i=1}^{k} Q_i \times H_i \times Q$, and (iii) for every $(p, h) \in Q_i \times H_i$ there exists exactly one transition $(p, h, q) \in \Delta$. For convenience, we represent Δ as a transition function $\delta \colon Q \times \Sigma \to Q$ instead. Observe that 1-tape (deterministic) automata are precisely NFAs (DFAs). A relation $R \subseteq (\Sigma^*)^k$ is *deterministic rational* if there exists a deterministic k-tape automaton $\mathcal{A}$ such that $R(\mathcal{A}) = \{(w_1 \dashv, \ldots, w_k \dashv) \mid (w_1, \ldots, w_k) \in R\}$ where $\dashv \notin \Sigma$ is a fresh endmarker.

c) Synchronous relations: Let $\bot \notin \Sigma$ be a fresh padding symbol. For finite words $w_1, \ldots, w_k \in \Sigma^*$ with $w_i = a_{i,1} \cdots a_{i,n_i}$ we define the convolution $w_1 \otimes \cdots \otimes w_k$ of length $n = \max\{n_1, \ldots, n_k\}$ by

$$w_1 \otimes \cdots \otimes w_k \stackrel{\text{def}}{=} \begin{bmatrix} w_1 \\ \vdots \\ w_k \end{bmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} a'_{1,1} \\ \vdots \\ a'_{k,1} \end{pmatrix} \cdots \begin{pmatrix} a'_{1,n} \\ \vdots \\ a'_{k,n} \end{pmatrix} \in ((\Sigma \cup \{\bot\})^k)^*$$

where $a'_{i,j} = a_{i,j}$ if $j \leq n_i$ and $a'_{i,j} = \bot$ otherwise. Similarly, we define the convolution for infinite words where the padding symbol $\bot$ is not needed. A relation R over (in)finite words is (ω-)*synchronous* if $\otimes R \stackrel{\text{def}}{=} \{w_1 \otimes \cdots \otimes w_k \mid (w_1, \ldots, w_k) \in R\}$ is a (ω-)regular language. A (ω-)synchronous relation R is always given by a finite (ω-)automaton for the language $\otimes R$.

d) Recognizable relations: A k-ary relation R on (in)finite words is (ω-)*recognizable* if it is a finite union $R = \bigcup_{i=1}^{n} L_{i,1} \times \cdots \times L_{i,k}$ of Cartesian products of (ω-)regular languages $L_{i,j}$.

Example 1. The relation $R_1 = \{(x, y) \mid |x| + |y| \geq 2\}$ over some finite alphabet Σ is recognizable, since it can be written as the union of the Cartesian products $\Sigma^{\geq 2} \times \Sigma^*$, $\Sigma^{\geq 1} \times \Sigma^{\geq 1}$, and $\Sigma^* \times \Sigma^{\geq 2}$ where $\Sigma^{\geq \ell}$ contains all words of length at least ℓ. The equality relation $R_2 = \{(x, x) \mid x \in \Sigma^*\}$ is synchronous but clearly not recognizable. The relation $R_3 = \{(x, y) \mid x$ is a scattered subword of $y\}$ is deterministic rational (the deterministic automaton greedily embeds x into y) but not synchronous. The relation $R_4 = \{(x, y) \mid x$ is an infix of $y\}$ is rational but not deterministic rational.

III. DECIDING RECOGNIZABILITY VIA FINITE-INDEX EQUIVALENCES

The key to decide recognizability of relations is a characterization by equivalence relations, akin to the Myhill-Nerode equivalence for languages. Let D be any domain (e.g. Σ^* or Σ^ω) and let $R \subseteq D^k$ be a k-ary relation. For $I \subseteq \{1, \ldots, k\}$, and two tuples $\boldsymbol{u} \in D^{|I|}$, $\boldsymbol{v} \in D^{k-|I|}$ we define $\boldsymbol{u} \odot_I \boldsymbol{v} \in D^k$ to be the unique k-tuple whose projection to I is $\boldsymbol{u}$ and whose projection to $\{1, \ldots, k\} \setminus I$ is $\boldsymbol{v}$. Define the equivalence relation $\approx_I^R$ on $D^{|I|}$ by

$$\boldsymbol{x} \approx_I^R \boldsymbol{y} \stackrel{\text{def}}{\iff} \text{for all } \boldsymbol{z} \in D^{k-|I|}:$$
$$(\boldsymbol{x} \odot_I \boldsymbol{z} \in R \iff \boldsymbol{y} \odot_I \boldsymbol{z} \in R).$$

Usually, R will be clear from the context and we simply write $\approx_I$. If $I = \{j\}$ is a singleton we also write $\odot_j$ and $\approx_j$ instead of $\odot_I$ and $\approx_I$. Notice that $\approx_{[1,j]}$ coincides with the relation $\sim_j$ from the introduction. For example, if R_1 is the relation from Example 1, then $\approx_1^{R_1}$ has three equivalence classes $\Sigma^{\geq 2}$, Σ and $\{\varepsilon\}$. We need the following characterization of recognizable relations.

Proposition 1. *Let $R \in$ **Rat** $\cup$ ω-**Sync**. The following are equivalent:*

1) *R is (ω-)recognizable.*
2) *$\approx_{[1,j]}^R$ has finite index for all $j \in [1, k-1]$.*
3) *$\approx_j^R$ has finite index for all $j \in [1, k-1]$.*

Equivalence of 1) and 2) was proved in [19, Proposition 3.8] for rational relations and in [29, Lemma 3] for ω-synchronous relations. In the following we prove equivalence of 2) and 3) by applying a result by Cosmadakis, Kuper, and Libkin [37]. We say that $R \subseteq D^k$ is an *I-relation* if there exists a relation $S \subseteq D^{|I|}$ such that $R = \{\boldsymbol{u} \odot_I \boldsymbol{v} \mid \boldsymbol{u} \in S, \; \boldsymbol{v} \in D^{k-|I|}\}$. Let P be a partition of $\{1, \ldots, k\}$. We say that R *conforms* to P if R is a finite Boolean combination of relations $R_1, \ldots, R_n$ where each R_i is an I-relation for some $I \in P$. For example, a relation R conforms to the *discrete partition* $\{\{1\}, \ldots, \{k\}\}$ if and only if R is a finite Boolean combination of Cartesian products $L_1 \times \cdots \times L_k$ of sets L_i. The following lemma is easy to show:

Lemma 1. *The equivalence relation $\approx_I^R$ has finite index if and only if R conforms to $\{I, [1, k] \setminus I\}$.*

If R conforms to P then clearly R conforms to any partition P' that is coarser than P. The *coarsest refinement* $P_1 \sqcap P_2$ of two partitions is the set of all nonempty intersections $I_1 \cap I_2$ where $I_1 \in P_1$, $I_2 \in P_2$.

Theorem 4 ([37]). *If $R \subseteq D^k$ conforms to two partitions P_1, P_2 then also to their coarsest refinement $P_1 \sqcap P_2$.*

The partition of $[1, k]$ *generated* by subsets $I_1, \ldots, I_n \subseteq [1, k]$ is the coarsest refinement $P_1 \sqcap \cdots \sqcap P_n$ of the partitions $P_j = \{I_j, [1, k] \setminus I_j\}$. For example, the discrete partition is clearly generated by the singleton sets $\{1\}, \ldots, \{k-1\}$. It is also generated by all intervals $[1, j]$ for $j \in [1, k-1]$.

Lemma 2. *If a partition P is generated by $I_1, \ldots, I_n \subseteq [1, k]$ then $R \subseteq D^k$ conforms to P if and only if $\approx_{I_j}^R$ has finite index for all $j \in [1, n]$.*

Proof. By Theorem 4, R conforms to P if and only if R conforms to $\{I_j, [1, k] \setminus I_j\}$ for each $j \in [1, n]$. By Lemma 1 this is equivalent to finite index of $\approx_{I_j}^R$ for all $j \in [1, n]$. $\square$

Choosing the discrete partition on $[1, k]$ as P in Lemma 2 we obtain the equivalence of 2) and 3) in Proposition 1.

IV. Deciding ω-Recognizability in ω-**Sync**

The goal of this section is to prove Theorem 1. The lower bounds are inherited from the finite-word case by padding, since recognizability of synchronous relations is PSPACE-complete (resp. NL-complete) if the relation is given by an NFA (resp. DFA) [30]. For the upper bounds we follow the same approach as in [30], [31] for the recognizability problem for synchronous relations. Given an (ω-)synchronous relation R the complements $\not\approx_j^R$ are again (ω-)synchronous. In fact, if R is given by a (non)deterministic automaton, then a nondeterministic automaton for $\not\approx_j^R$ can be computed in logspace (polynomial space): Observe that $x \not\approx_j^R y$ if and only if

$$\exists \boldsymbol{z} \in (\Sigma^\omega)^{k-1} \colon (x \odot_j \boldsymbol{z} \in R \ \wedge \ y \odot_j \boldsymbol{z} \notin R) \ \vee$$
$$(x \odot_j \boldsymbol{z} \notin R \ \wedge \ y \odot_j \boldsymbol{z} \in R).$$

If R is given by a DPA $\mathcal{B}$, we can construct an DPA for $(\Sigma^\omega)^k \setminus R$ in logarithmic space and convert it into an NBA $\bar{\mathcal{B}}$ [38]. If R is given by an NBA $\mathcal{B}$ then this step incurs an exponential blowup but can still be done in polynomial space [39]. From $\mathcal{B}$ and $\bar{\mathcal{B}}$ we can construct NBAs for the relations $\not\approx_j$ in logspace (intersections, unions, and projections of NBAs are logspace computable).

By Proposition 1 it remains to check whether for some $j < k$ the relation $\not\approx_j^R$ has an *infinite clique*, i.e. an infinite sequence of pairwise distinct words $w_1, w_2, \ldots$ such that $w_{i_1} \not\approx_j^R w_{i_2}$ for all $i_1 < i_2$. In [31] it is shown that the infinite clique problem can be solved in nondeterministic logspace for arbitrary synchronous relations over finite words. For arbitrary ω-synchronous relations, it is a longstanding open problem

whether the infinite clique problem is decidable. However, in [29] it is shown to be decidable in double exponential time for ω-synchronous *co-equivalence relations*, i.e. complements of equivalence relations. In the following we will show that for those relations the infinite clique problem can even be solved in nondeterministic logspace. Applying this result to the relations $\not\approx_j^R$ yields an NL respectively PSPACE algorithm for ω-recognizability of ω-synchronous relations depending on whether R is given by a DPA or NBA.

Theorem 5. *It is NL-complete to decide, given a nondeterministic Büchi automaton for an ω-synchronous co-equivalence relation $\bar{E}$, whether $\bar{E}$ has an infinite clique.*

The rest of this section is devoted to proving Theorem 5. One challenge in finding infinite cliques in ω-synchronous relations is how to even finitely represent infinite clique, i.e. an infinite sequence of infinite words. A strong indicator that this is indeed difficult is that there are ω-synchronous relations which have infinite cliques but no *regular* infinite clique. One such example is the complement of the *equal ends* relation $\sim_e$ on Σ^ω where $u \sim_e v$ if and only if there exist $x, y \in \Sigma^*$, $z \in \Sigma^\omega$ with $|x| = |y|$ and $u = xz$ and $v = yz$. Kuske and Lohrey observed that, although $\not\sim_e$ has infinite cliques it does not have regular infinite cliques [40, Example 2.1].

Instead of checking whether $\bar{E}$ has an infinite clique, we follow the approach of Löding and Spinrath [29] and equivalently decide whether $\bar{E}$ has *unbounded cliques*: For each $n \geq 1$ there exists a clique $(w_1, \ldots, w_n)$ of size n in R, i.e. $(w_i, w_j) \in \bar{E}$ for all $1 \leq i < j \leq n$. The important observation made in [29] is that it suffices to search for unbounded cliques consisting of ultimately periodic words. Since ultimately periodic words uv^ω can be encoded by the finite word $u\#v$, this allows us to reduce the infinite clique problem over ω-synchronous co-equivalence relations to a question over synchronous relations over finite words.

For the rest of this section fix an NBA $\mathcal{A} = (Q, \Sigma^2, q_0, \Delta, F)$ for the complement $\bar{E} \subseteq \Sigma^\omega \times \Sigma^\omega$ of an equivalence relation E. Let us recall the results from [29] that will be used for our proof. Define the equivalence relation $E_\# \subseteq (\Sigma^* \# \Sigma^*)^2$ where $\# \notin \Sigma$ by

$$E_\# \stackrel{\text{def}}{=} \{(u\#v, x\#y) \mid (uv^\omega, xy^\omega) \in E, \ |u| = |x|, \ |v| = |y|\},$$

which is an ω-synchronous relation [29]. Let $L_\#(E)$ be a regular set of representatives of $E_\#$, e.g. consisting of the length-lexicographically minimal elements in each class [19, Proof of Proposition 3.9]. A language $L \subseteq \Sigma^*$ is *slender* if there exists a $k \in \mathbb{N}$ such that for all $\ell \in \mathbb{N}$ it holds that $|L \cap \Sigma^\ell| < k$. The following lemma is shown in [29, Lemmas 12 and 13].

Lemma 3. *One can write $L_\#(E) = \bigcup_{(i,j) \in I} P_i\{\#\}S_j$ for a finite index set $I \subseteq \mathbb{N}^2$ and non-empty regular languages $P_i, S_j \subseteq \Sigma^*$ for all $(i, j) \in I$. Furthermore, E has finite index if and only if P_i and S_j are slender for all $(i, j) \in I$.*

The approach in [29] for checking whether E has finite index is to construct automata for each of the languages P_i

and S_j and check whether they are slender. However, an automaton for the set of representatives $L_\#(E)$ might be double exponentially large, since its size is exponential in the size of the automaton for $E_\#$, which in turn is exponential in the size of the automaton for $\bar{E}$ via a construction using transition profiles. This results in a double exponential time algorithm given an automaton for $\bar{E}$. We deviate from this approach and use the slenderness property of the languages P_i and S_j only to identify the shape of unbounded cliques in $\bar{E}$. In a second step, we search for patterns in the automaton for $\bar{E}$ that witness unbounded cliques in $\bar{E}$. The existence of these patterns can be checked in nondeterministic logspace given an automaton for $\bar{E}$.

Lemma 4. *A regular language $L \subseteq \Sigma^*$ is not slender if and only if there are words $u, v, w, x, y \in \Sigma^*$ with $|v| = |w| = |x| > 0$ and $v \neq w$ such that $uv^*wx^*y \subseteq L$.*

Proof. Consider the minimal trimmed DFA for L. By [41, Theorem 4.23] L is not slender if and only if the DFA contains two distinct nonempty cycles $p \xrightarrow{v} p$, $q \xrightarrow{x} q$ (distinct means that their set of transitions are distinct), and a run $p \xrightarrow{w} q$. In particular, uv^*wx^*y is contained in L where u and y are words read from the initial state to p, and from q to some final state. We can ensure that $|w| \leq |v| = |x|$ by replacing v and x by $v^{k|x|}$ and $x^{k|v|}$, respectively, for a sufficiently large number k. Furthermore, we can ensure $|v| = |w| = |x|$ by extending the run $p \xrightarrow{w} q$ to $p \xrightarrow{wx_1} r$ for some prefix x_1 of x with $|wx_1| = |x|$ and rebasing the cycle $q \xrightarrow{x} q$ on state r. $\square$

The following lemma distinguishes two types of cliques: On the one hand, there are cliques whose words differ in a finite prefix but have equal ends; on the other hand, there are cliques whose words do not have equal ends. For example, consider the equality relation $=$. In the complement $\neq$ we can find the cliques $(a^i b^{n-i} a^\omega)_{0 \leq i \leq n}$ for all $n \in \mathbb{N}$ of words that only differ in a finite prefix. On the other hand, if we consider the equal ends $\sim_e$ equivalence relation then we observe that it does not suffice to look at the finite prefixes of words to determine whether they are in relation or not. In the complement $\not\sim_e$ we can find the cliques $((a^i b^{n-i})^\omega)_{0 \leq i \leq n}$ for all $n \in \mathbb{N}$ of words that differ in the periodic part.

Lemma 5. *$\bar{E}$ contains an infinite clique if and only if $\bar{E}$ contains cliques of the form $(uv^i wx^{n-i} yz^\omega)_{0 \leq i \leq n}$ or $(z(uv^i wx^{n-i}y)^\omega)_{0 \leq i \leq n}$ for all $n \in \mathbb{N}$ where $u, v, w, x, y, z \in \Sigma^*$ with $|v| = |w| = |x| > 0$ and $v \neq w$.*

Proof. The "if" direction is immediate since the existence of such cliques imply that $\bar{E}$ contains unbounded cliques and therefore also an infinite clique.

For the "only if" direction assume that $\bar{E}$ contains an infinite clique which means that E has infinite index. Then by Lemma 3 there are non-empty regular languages $P, S \subseteq \Sigma^*$ with $P\{\#\}S \subseteq L_\#(E)$ such that P or S is not slender. By Lemma 4 there are $u, v, w, x, y \in \Sigma^*$ with $|v| = |w| = |x| > 0$ and $v \neq w$ such that $uv^*wx^*y \subseteq P$ or $uv^*wx^*y \subseteq S$. If $uv^*wx^*y \subseteq P$ we pick a word $z \in S$ from the non-

empty language S. Then $uv^*wx^*y\#z \subseteq L_\#(E)$. Since all words of the form $uv^i wx^j y$ are pairwise different, $\bar{E}$ contains the clique $(uv^i wx^{n-i} yz^\omega)_{0 \leq i \leq n}$ for all $n \in \mathbb{N}$. Similarly, if $uv^*wx^*y \subseteq S$, we pick a word $z \in P$ and find the cliques $(z(uv^i wx^{n-i}y)^\omega)_{0 \leq i \leq n}$ of size n in $\bar{E}$. $\square$

A *3-cycles pattern* consists of states $q_1, q_2, q_3, q_4, q_5 \in Q$ and words $u, v, w, x, y \in \Sigma^*$ with $|v| = |w| = |x| > 0$ and $v \neq w$ such that

$$q_1 \xrightarrow{\left[\begin{smallmatrix}u\\u\end{smallmatrix}\right]} q_2, \quad q_2 \xrightarrow{\left[\begin{smallmatrix}v\\v\end{smallmatrix}\right]} q_2, \quad q_2 \xrightarrow{\left[\begin{smallmatrix}w\\v\end{smallmatrix}\right]} q_3, \quad q_3 \xrightarrow{\left[\begin{smallmatrix}x\\v\end{smallmatrix}\right]} q_3,$$

$$q_3 \xrightarrow{\left[\begin{smallmatrix}x\\w\end{smallmatrix}\right]} q_4, \quad q_4 \xrightarrow{\left[\begin{smallmatrix}x\\x\end{smallmatrix}\right]} q_4, \quad q_4 \xrightarrow{\left[\begin{smallmatrix}y\\y\end{smallmatrix}\right]} q_5.$$

We say that the above is a *3-cycles pattern from q_1 to q_5*. The 3-cycles pattern is called *final* if one of the runs $q_1 \xrightarrow{\left[\begin{smallmatrix}u\\u\end{smallmatrix}\right]} q_2$, $q_2 \xrightarrow{\left[\begin{smallmatrix}w\\v\end{smallmatrix}\right]} q_3$, $q_3 \xrightarrow{\left[\begin{smallmatrix}x\\w\end{smallmatrix}\right]} q_4$, $q_4 \xrightarrow{\left[\begin{smallmatrix}y\\y\end{smallmatrix}\right]} q_5$ visits a final state. If there exists a (final) 3-cycles pattern from p to q we write $q_1 \xrightarrow{\text{3CP}} q_5$ and $q_1 \xrightarrow{\text{3CP}}_F q_5$, respectively.

Clearly $p \xrightarrow{\text{3CP}} q$ implies that the automaton contains p-q-runs reading $uv^i wx^{n-i} y \otimes uv^j wx^{n-j} y$ for all $i < j \leq n$, for some u, v, w, x, y with $|v| = |w| = |x| > 0$ and $v \neq w$. To prove that the converse also holds, we use transition profiles. A *transition profile* $\tau = (\Rightarrow, \xRightarrow{F})$ over $\mathcal{A}$ consists of two binary relations $\Rightarrow, \xRightarrow{F}$ over Q. For each word $w \in (\Sigma^2)^*$ we define the transition profile $\tau(w)$ such that $p \Rightarrow q$ if and only if there exists a run $p \xrightarrow{w} q$, and $p \xRightarrow{F} q$ if and only if there exists a run $p \xrightarrow{w} q$ visiting a final state. It is easy to see that $\tau(uv)$ is determined by $\tau(u)$ and $\tau(v)$, and therefore the set $\mathsf{TP}(\mathcal{A}) = \{\tau(w) \mid w \in (\Sigma^2)^*\}$ forms a finite monoid with the well-defined operation $\tau(u) \cdot \tau(v) = \tau(uv)$ and neutral element $\tau(\varepsilon)$. An element s in a monoid M is *idempotent* if $s^2 = s$. Every finite monoid M has an *idempotent exponent*, i.e. a number $n \geq 1$ so that s^n is idempotent for all $s \in M$.

Lemma 6. *Let $p \in \mathbb{N}$ be the idempotent exponent of $\mathsf{TP}(\mathcal{A})$. If for words $u, v, w, x, y, z \in \Sigma^*$ with $|v| = |w| = |x| > 0$ and $v \neq w$ and states $q_1, q_5 \in Q$ there exists a run ρ in $\mathcal{A}$ from q_1 to q_5 reading*

$$\left[\begin{smallmatrix}u\\u\end{smallmatrix}\right]\left[\begin{smallmatrix}v^{pn}\\v^{pn}\end{smallmatrix}\right]\left[\begin{smallmatrix}v^{p-1}w\\v^p\end{smallmatrix}\right]\left[\begin{smallmatrix}x^{pn}\\v^{pn}\end{smallmatrix}\right]\left[\begin{smallmatrix}x^p\\v^{p-1}w\end{smallmatrix}\right]\left[\begin{smallmatrix}x^{pn}\\x^{pn}\end{smallmatrix}\right]\left[\begin{smallmatrix}y\\y\end{smallmatrix}\right]$$

then $q_1 \xrightarrow{\text{3CP}} q_5$. If ρ visits a final state then $q_1 \xrightarrow{\text{3CP}}_F q_5$.

Proof. We will use the fact that $\left[\begin{smallmatrix}v^p\\v^p\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}x^p\\v^p\end{smallmatrix}\right]$, and $\left[\begin{smallmatrix}x^p\\x^p\end{smallmatrix}\right]$ are idempotent in $\mathsf{TP}(\mathcal{A})$. Let us replace v by v^p, w by $v^{p-1}w$, and x by x^p. Now the words $\left[\begin{smallmatrix}v\\v\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}x\\v\end{smallmatrix}\right]$, and $\left[\begin{smallmatrix}x\\x\end{smallmatrix}\right]$ are idempotent in $\mathsf{TP}(\mathcal{A})$, and ρ becomes a run reading

$$\left[\begin{smallmatrix}u\\u\end{smallmatrix}\right]\left[\begin{smallmatrix}v^n\\v^n\end{smallmatrix}\right]\left[\begin{smallmatrix}w\\v\end{smallmatrix}\right]\left[\begin{smallmatrix}x^n\\v^n\end{smallmatrix}\right]\left[\begin{smallmatrix}x\\w\end{smallmatrix}\right]\left[\begin{smallmatrix}x^n\\x^n\end{smallmatrix}\right]\left[\begin{smallmatrix}y\\y\end{smallmatrix}\right]. \tag{1}$$

The sequence of $n + 1$ states visited before and after reading each of the n factors $\left[\begin{smallmatrix}v\\v\end{smallmatrix}\right]$ must contain a repeated state q_2,

and similarly for the factors $\begin{bmatrix} x \\ v \end{bmatrix}$ and $\begin{bmatrix} v \\ v \end{bmatrix}$. Therefore we find intermediate states q_2, q_3, q_4 so that ρ has the form

$$\rho_1: q_1 \xrightarrow{\begin{bmatrix} uv^{i_1} \\ uv^{i_1} \end{bmatrix}} q_2, \qquad \sigma_2: q_2 \xrightarrow{\begin{bmatrix} v^{i_2} \\ v^{i_2} \end{bmatrix}} q_2,$$

$$\rho_2: q_2 \xrightarrow{\begin{bmatrix} v^{i_3} \\ v^{i_3} \end{bmatrix}\begin{bmatrix} w \\ v \end{bmatrix}\begin{bmatrix} x^{j_1} \\ v^{j_1} \end{bmatrix}} q_3, \qquad \sigma_3: q_3 \xrightarrow{\begin{bmatrix} x^{j_2} \\ v^{j_2} \end{bmatrix}} q_3$$

$$\rho_3: q_3 \xrightarrow{\begin{bmatrix} x^{j_3} \\ v^{j_3} \end{bmatrix}\begin{bmatrix} x \\ w \end{bmatrix}\begin{bmatrix} x^{k_1} \\ x^{k_1} \end{bmatrix}} q_4, \quad \sigma_4: q_4 \xrightarrow{\begin{bmatrix} x^{k_2} \\ x^{k_2} \end{bmatrix}} q_4, \qquad (2)$$

$$\rho_4: q_4 \xrightarrow{\begin{bmatrix} x^{k_3}y \\ x^{k_3}y \end{bmatrix}} q_5$$

for some numbers $i_1, j_1, k_1, i_3, j_3, k_3 \geq 0$ and $i_2, j_2, k_2 \geq 1$. Since $\begin{bmatrix} v \\ v \end{bmatrix}$, $\begin{bmatrix} x \\ v \end{bmatrix}$, and $\begin{bmatrix} x \\ x \end{bmatrix}$ are idempotent in $\mathsf{TP}(\mathcal{A})$, there exist runs $\tilde{\rho}_1, \tilde{\sigma}_2, \tilde{\rho}_2, \tilde{\sigma}_3, \tilde{\rho}_3, \tilde{\sigma}_4, \tilde{\rho}_4$ as in Equation (2) for $i_\ell = j_\ell = k_\ell = 1$ for all $\ell \in [1,3]$. Then the five words uv, v^3, vwx, x^3, xy form the required 3-cycles pattern from q_1 to q_5.

Assume that ρ visits a final state. We can ensure that the final state occurs in one of the subruns ρ_i in Equation (2): If the final state occurs in one of the cycles σ_i then we can append the cycle σ_i to ρ_i. By the $\xrightarrow{F}$-component of transition profiles we can then choose the run $\tilde{\rho}_i$ to visit a final state again, and therefore $q_1 \xrightarrow{3\mathrm{CP}}_F q_5$. $\qquad\square$

The next lemma shows that a 3-cycles pattern can be used to detect unbounded cliques in $\bar{E}$ where the words differ in the finite prefix.

Lemma 7. $\bar{E}$ *contains cliques* $(uv^i wx^{n-i} yz^\omega)_{0 \leq i \leq n}$ *for all* $n \in \mathbb{N}$ *where* $u, v, w, x, y, z \in \Sigma^*$ *with* $|v| = |w| = |x| > 0$ *and* $v \neq w$ *if and only if there is a 3-cycles pattern in $\mathcal{A}$ from* q_0 *to some state* $q \in Q$ *such that* $\begin{bmatrix} z'^\omega \\ z'^\omega \end{bmatrix}$ *is accepted from* q *for some word* $z' \in \Sigma^*$.

Proof. We first observe that $\bar{E}$ contains cliques $(uv^i wx^{n-i} yz^\omega)_{0 \leq i \leq n}$ as on the LHS of the lemma if and only if

$$\begin{bmatrix} u \\ u \end{bmatrix}\begin{bmatrix} v \\ v \end{bmatrix}^*\begin{bmatrix} w \\ v \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix}^*\begin{bmatrix} x \\ w \end{bmatrix}\begin{bmatrix} x \\ x \end{bmatrix}^*\begin{bmatrix} y \\ y \end{bmatrix}\begin{bmatrix} z \\ z \end{bmatrix}^\omega \subseteq L(\mathcal{A}) \qquad (3)$$

for some $u, v, w, x, y, z \in \Sigma^*$ with $|v| = |w| = |x| > 0$ and $v \neq w$. Then the "if" direction of the lemma follows directly. For the "only if" direction assume that Equation (3) holds. Let $n \stackrel{\text{def}}{=} |Q|$ and $p \in \mathbb{N}$ be the idempotent exponent of $\mathsf{TP}(\mathcal{A})$. Then there is a run of $\mathcal{A}$ on $\begin{bmatrix} u(v^p)^n v^{p-1} w(x^p)^{2n+1} y \\ u(v^p)^{2n+1} v^{p-1} w(x^p)^n y \end{bmatrix}$ from q_0 to some state $q \in Q$ such that $\begin{bmatrix} z^\omega \\ z^\omega \end{bmatrix}$ is accepted from q. Applying Lemma 6 yields the desired 3-cycles pattern. $\qquad\square$

The following lemma shows which pattern occurs if the words differ in the periodic part. In Lemma 9 we will see that this pattern is also sufficient to show the existence of unbounded cliques.

Lemma 8. *If* $\bar{E}$ *contains cliques* $(z(uv^i wx^{n-i} y)^\omega)_{0 \leq i \leq n}$ *for all* $n \in \mathbb{N}$ *where* $u, v, w, x, y, z \in \Sigma^*$ *with* $|v| = |w| = |x| > 0$ *and* $v \neq w$, *then there are states* $q_1, \ldots, q_\ell \in Q$ *such that*

- $q_0 \xrightarrow{\begin{bmatrix} z \\ z \end{bmatrix}} q_1,$

- $q_1 \xrightarrow{3\mathrm{CP}} q_2 \xrightarrow{3\mathrm{CP}} q_3 \xrightarrow{3\mathrm{CP}} \cdots \xrightarrow{3\mathrm{CP}} q_{\ell-1} \xrightarrow{3\mathrm{CP}}_F q_\ell,$
- $q_k = q_\ell$ *for some* $k < \ell$.

Proof. Suppose that $\bar{E}$ contains cliques $(z(uv^i wx^{n-i} y)^\omega)_{0 \leq i \leq n}$ with $|v| = |w| = |x| > 0$ and $v \neq w$. Let t be the word from Equation (1). Since $\begin{bmatrix} z \\ z \end{bmatrix} t^\omega$ is accepted by $\mathcal{A}$, it has an accepting run of the form

$$q_0 \xrightarrow{\begin{bmatrix} z \\ z \end{bmatrix}} q_1 \xrightarrow{t} q_2 \xrightarrow{t} q_3 \xrightarrow{t} \cdots .$$

Let $m \in \mathbb{N}$ such that $\{q_0, \ldots, q_m\} = \{q_i \mid i \in \mathbb{N}\}$, i.e. all states q_i have been visited at least once after reaching q_m. Since the run visits some final state infinitely often, there exists $\ell > m$ such that the subrun between $q_{\ell-1}$ and q_ℓ visits a final state. Furthermore, there exists $k \leq m$ such that $q_k = q_\ell$. $\qquad\square$

Lemma 9. *If there are states* $q_1, \ldots, q_\ell \in Q$ *such that*

- $q_0 \xrightarrow{\begin{bmatrix} z \\ z \end{bmatrix}} q_1,$
- $q_1 \xrightarrow{3\mathrm{CP}} q_2 \xrightarrow{3\mathrm{CP}} q_3 \xrightarrow{3\mathrm{CP}} \cdots \xrightarrow{3\mathrm{CP}} q_{\ell-1} \xrightarrow{3\mathrm{CP}}_F q_\ell,$
- $q_k = q_\ell$ *for some* $k < \ell$,

then $\bar{E}$ *contains unbounded cliques.*

Proof. Let $u_j, v_j, w_j, x_j, y_j \in \Sigma^*$ be the words of the 3-cycles pattern from q_j to q_{j+1} for $j \in [1, \ell-1]$. Define $t_j(i, n) \stackrel{\text{def}}{=} u_j v_j^i w_j x_j^{n-i} y_j$ for all $0 \leq i \leq n$ and $1 \leq j < \ell$. Then

$$\big(t_1(i,n) \cdots t_{k-1}(i,n)\big(t_k(i,n) \cdots t_\ell(i,n)\big)^\omega\big)_{0 \leq i \leq n}$$

forms a clique in $\bar{E}$ for each $n \in \mathbb{N}$. $\qquad\square$

We are now ready to prove Theorem 5. Since $\bar{E}$ has an infinite clique if and only if it has unbounded cliques, by Lemmas 5 and 7 to 9 it suffices to check whether $\mathcal{A}$ contains the pattern in Lemma 7 or the pattern in Lemmas 8 and 9.

First, we can check in NL whether, given states $p, q \in Q$, there exists a word $z \in \Sigma^*$ with $p \xrightarrow{\begin{bmatrix} z \\ z \end{bmatrix}} q$, and whether there exists a word $z \in \Sigma^*$ such that $\begin{bmatrix} z^\omega \\ z^\omega \end{bmatrix}$ is accepted from q. Furthermore, given two states $q_1, q_5 \in Q$, we can check whether $q_1 \xrightarrow{3\mathrm{CP}} q_5$ in NL as follows: Construct in logspace an NFA $\mathcal{A}_{q_1,q_5}$ which reads a convolution $u \otimes v \otimes w \otimes x \otimes y$ with $|v| = |w| = |x|$ and $v \neq w$. It initially guesses and stores the states q_2, q_3, q_4. Then, it simulates 7 copies of $\mathcal{A}$ in parallel to check the existence of the runs as in the definition of a 3-cycles pattern. Observe that $q_1 \xrightarrow{3\mathrm{CP}} q_5$ if and only if $\mathcal{A}_{q_1,q_5}$ accepts some word, hence, it suffices to check nonemptiness of $\mathcal{A}_{q_1,q_5}$ in NL. Similarly, we can test $q_1 \xrightarrow{3\mathrm{CP}}_F q_5$ in NL. This allows us to detect the pattern in Lemma 7 and the pattern in Lemmas 8 and 9 in NL.

NL-hardness follows by a reduction from the finite word case [31] by padding.

Application to monadic decomposability over ω-automatic structures: Recognizability of relations over words is closely connected to the notion of monadic decomposition. A formula $\varphi(x_1, \ldots, x_k)$ is *monadically decomposable* over a logical structure $\mathfrak{A}$ if φ is equivalent to a Boolean combination of monadic formulas, i.e. formulas with a single free variable [42]. While generally undecidable, Libkin provides

sufficient conditions on $\mathfrak{A}$ under which the problem of testing monadic decomposability becomes decidable [43, Theorem 3]. Under these conditions a formula φ defining a k-ary relation R is monadically decomposable if and only if $\approx_{[1,j]}^R$ has finite index for all $j < k$ [43, Lemma 4]. In its generality the algorithm in [43] is not very efficient since it uses an unstructured enumeration procedure to find so called *definable invariant Skolem functions*.

There is a more straightforward procedure for monadic decomposability if $\mathfrak{A}$ is ω-automatic [24], i.e. its domain and relations are given by ω-synchronous automata. In this setting we can translate φ into an ω-synchronous automaton for the relation R over infinite words defined by φ. Then, we construct automata for the relations $\napprox_{[1,j]}^R$ and solve the infinite clique problem by Theorem 5. In fact, if φ is quantifier-free (this assumption is also made in [42]), the automata for $\napprox_{[1,j]}^R$ are constructible in polynomial space. Combined with the NL-algorithm for the infinite clique problem, this yields PSPACE-complexity for testing monadic decomposability.

An example for an ω-automatic structure that satisfies the conditions of [43, Theorem 3] is real linear arithmetic (RLA) $(\mathbb{R}; +, <, 0, 1)$. Its extension $(\mathbb{R}; \mathbb{Z}, +, <, 0, 1)$ to mixed real-integer linear arithmetic (RILA) is still ω-automatic [44], but it is not immediately clear whether it fulfills the conditions of [43, Theorem 3]. However, we can use the fact that in the (standard) ω-automatic presentation of RILA ultimately periodic words are rational numbers and therefore definable in RILA.

Lemma 10. *Let $\mathfrak{A}$ be an ω-automatic structure with an ω-automatic presentation over alphabet Σ such that each ultimately periodic word over Σ that represents an element in the domain is definable in $\mathfrak{A}$. Then a formula φ in $\mathfrak{A}$ defining the relation $R \subseteq (\Sigma^\omega)^k$ is monadically decomposable if and only if $\approx_j^R$ has finite index for all $j \in [1, k]$.*

Proof. The "only if" direction is clear. For the "if" direction assume that $\approx_j^R$ has finite index for all $j \in [1, k]$. It was observed in [29, Proof of Lemma 3] that every finite-index ω-synchronous equivalence relation has a set of ultimately periodic representatives. Let $A_1, \ldots, A_k$ be such representative sets for $\approx_1^R, \ldots, \approx_k^R$. Now, $\varphi(x_1, \ldots, x_k)$ is equivalent to the following Boolean combination of monadic formulas:

$$\bigvee_{(a_1, \ldots, a_k) \in R \cap (A_1 \times \cdots \times A_k)} \bigwedge_{j=1}^k x_j \approx_j a_j$$

The statement $x_j \approx_j a_j$ is definable since $\approx_j$ is definable and the ultimately periodic words in A_j are definable in $\mathfrak{A}$. $\qquad\square$

Therefore, under the assumption in Lemma 10 (e.g. for RILA) we can use the same approach from above to decide monadic decomposability for quantifier-free formulas in polynomial space.

V. DECIDING RECOGNIZABILITY IN DRat

In this section we will show how to test whether a deterministic rational relation is recognizable (Theorem 2) and,

if so, how to construct an equivalent independent multitape automaton. Notice that we can ignore the endmarkers in the definition of **DRat** since a relation R is recognizable if and only if $\{\boldsymbol{w}(\dashv, \ldots, \dashv) \mid \boldsymbol{w} \in R\}$ is recognizable. Hence, for the rest of this section let $R \subseteq (\Sigma^*)^k$ with $R = R(\mathcal{A})$ for some deterministic k-tape automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ with n states. Furthermore we assume that all states are reachable from q_0. We also write R_q for the relation recognized from state q, i.e. $R_q \overset{\text{def}}{=} R(\mathcal{A}_q)$ where $\mathcal{A}_q = (Q, \Sigma, q, \delta, F)$.

A. Witness for nonrecognizability

To decide whether R is recognizable it suffices to check whether the equivalence relations $\approx_1, \ldots, \approx_{k-1}$ have finite index by Proposition 1. To keep notation clean in the following we will focus on how to test whether $\approx_1$ has finite index. By permuting the components of R we can reduce testing finite-index of any $\approx_j$ to the $\approx_1$-case.

We provide equivalent characterizations (Proposition 2) of when $\approx_1$ has infinite index, which will be used for the decision procedures in Theorem 2. The characterizations will be deduced from the proof of Lemma 3.5 in [19], which states that, if $\approx_1$ has finite index, then any word is $\approx_1$-equivalent to a word whose length is exponentially bounded in n. We need a few definitions from [19]. A nonempty word $v_1 \in \Sigma^+$ is *null-transparent* if for all $s, t \in Q_1$ we have $s \xrightarrow{(v_1, \varepsilon)} t$ implies $t \xrightarrow{(v_1, \varepsilon)} t$. In other words, v_1 induces an *idempotent* transformation on Q_1. Since every element m in a finite monoid has an idempotent power m^ℓ, every non-empty word v_1 has a null-transparent power v_1^ℓ. We call a run $s \xrightarrow{(x, z)} t$ an *N-path* if the run switches from Q_1 to $Q \setminus Q_1$ at most N times. A nonempty word $y \in \Sigma^+$ is called *N-invisible in the context of $x \in \Sigma^*$* if any N-path $s \xrightarrow{(x, z)} t$ with $t \in Q_1$ implies $t \xrightarrow{(y, \varepsilon)} t$.

Lemma 11 ([19, Lemma 3.4]). *Let n be the number of states of $\mathcal{A}$ and let $u_1 \cdots u_\ell \in \Sigma^*$ be a product of ℓ nonempty words.*
1) *If $\ell > n!$ then some factor $u_{i+1} \cdots u_j$ is null-transparent.*
2) *If $\ell > 2(Nn)^N$ then some factor $u_{i+1} \cdots u_j$ is N-invisible in the context of $u_1 \ldots u_i$.*

We say that a set S *separates* two sets X and Y if $X \subseteq S$ and $Y \cap S = \emptyset$, or $Y \subseteq S$ and $X \cap S = \emptyset$. If X is a singleton $\{x\}$ we also say that S separates x and Y (similarly for Y).

Proposition 2. *The following conditions are equivalent:*
1) *$\approx_1$ has infinite index.*
2) *There exist words $x, y, z \in \Sigma^*$ such that y is $nn!$-invisible in the context of x and $xyz \napprox_1 xz$.*
3) *There exist $\boldsymbol{v}, \boldsymbol{w} \in (\Sigma^*)^k$ and a state $q \in Q$ such that $q \xrightarrow{\boldsymbol{v}} q$, v_1 is null-transparent, R_q separates $\boldsymbol{w}$ and $(v_1, \varepsilon)\boldsymbol{w}$.*
4) *There exist $\boldsymbol{v}, \boldsymbol{w} \in (\Sigma^*)^k$ and a state $q \in Q$ such that $q \xrightarrow{\boldsymbol{v}} q$, and R_q separates $\boldsymbol{w}$ and $(v_1, \varepsilon)^+\boldsymbol{w}$.*

The implication $(2 \Rightarrow 1)$ already appeared in [19, Proof of Lemma 3.5]. In our understanding, to prove this implication

the authors used 3) as an intermediate step. Unfortunately, the proof for the implication (3 ⇒ 1) contains an argument that we could not follow, see Appendix A for a discussion. For completeness, we reprove the implication (3 ⇒ 1) using 4) as an intermediate step.

Proof of Proposition 2. Let us start with the easy directions.

(4 ⇒ 1): Consider any run $q_0 \xrightarrow{u} q$. Then $u_1 v_1^i w_1 \not\approx_1 u_1 v_1^{i+j} w_1$ for all $i \geq 0$, $j \geq 1$ because R separates $\boldsymbol{uv}^i \boldsymbol{w}$ and $\boldsymbol{uv}^i (v_1, \boldsymbol{\varepsilon})^j \boldsymbol{w}$. Hence $\approx_1$ has infinite index.

(1 ⇒ 2): Assume that 2) is false. By Lemma 11, any word of length at least $f(nn!)$ where $f(N) \stackrel{\text{def}}{=} 2(Nn)^n$ can be written as uvw where v is nonempty and $nn!$-invisible in the context of u, and therefore $uvw \approx_1 uw$. By repeating this argument, we obtain for any word an $\approx_1$-equivalent word of length at most $f(nn!)$. Therefore, $\approx_1$ has finite index.

(2 ⇒ 3): Assume that $xyz \not\approx_1 xz$ where y is $nn!$-invisible in the context of x. Choose a length-minimal tuple $\boldsymbol{t} \in (\Sigma^*)^{k-1}$ such that

$$(xyz, \boldsymbol{t}) \in R \iff (xz, \boldsymbol{t}) \notin R. \quad (4)$$

Let ρ be a prefix of the run on $(xyz, \boldsymbol{t})$ which reads x on the first tape. Observe that ρ is not a $nn!$-path since y is $nn!$-invisible in the context of x and otherwise one could remove the $(y, \boldsymbol{\varepsilon})$-loop from the run, which would contradict Equation (4). In particular, ρ reads at least $nn!$ symbols from $\boldsymbol{t}$. Consider the sequence of states in ρ visited after reading a symbol from $\boldsymbol{t}$. There is a state q which is visited more than $n!$ times. We can factor $x = \alpha_1 \cdots \alpha_{\ell+1}$ and a prefix of $\boldsymbol{t}$ into nonempty words $\tau_1 \cdots \tau_{\ell+1}$ such that

$$q_0 \xrightarrow{(\alpha_1, \tau_1)} q \xrightarrow{(\alpha_2, \tau_2)} q \xrightarrow{(\alpha_3, \tau_3)} \ldots \xrightarrow{(\alpha_\ell, \tau_\ell)} q \xrightarrow{(\alpha_{\ell+1}, \tau_{\ell+1})} p$$

and $\ell > n!$. By Lemma 11 there exists a null-transparent factor $\alpha_{i+1} \cdots \alpha_j$ for some $1 \leq i < j \leq \ell$. Let us set $x_1 = \alpha_1 \cdots \alpha_i$, $x_2 = \alpha_{i+1} \cdots \alpha_j$, and $x_3 = \alpha_{j+1} \cdots \alpha_{\ell+1}$. Consider the corresponding decomposition $\boldsymbol{t} = \boldsymbol{t}_1 \boldsymbol{t}_2 \boldsymbol{t}_3$ such that

$$q_0 \xrightarrow{(x_1, \boldsymbol{t}_1)} q \xrightarrow{(x_2, \boldsymbol{t}_2)} q \xrightarrow{(x_3yz, \boldsymbol{t}_3)} r_+ \quad (5)$$

and

$$q_0 \xrightarrow{(x_1, \boldsymbol{t}_1)} q \xrightarrow{(x_2, \boldsymbol{t}_2)} q \xrightarrow{(x_3z, \boldsymbol{t}_3)} r_- \quad (6)$$

where exactly one of the states r_+, r_- belongs to F.

Since $\boldsymbol{t}$ is a length-minimal tuple satisfying Equation (4) and $\boldsymbol{t}_2$ is nonempty we know that

$$(xyz, \boldsymbol{t}_1 \boldsymbol{t}_3) \in R \iff (xz, \boldsymbol{t}_1 \boldsymbol{t}_3) \in R$$

and thus

$$(x_2 x_3 yz, \boldsymbol{t}_3) \in R_q \iff (x_2 x_3 z, \boldsymbol{t}_3) \in R_q. \quad (7)$$

We claim that either (i) R_q separates $(x_2 x_3 yz, \boldsymbol{t}_3)$ and $(x_3 yz, \boldsymbol{t}_3)$ or (ii) R_q separates $(x_2 x_3 z, \boldsymbol{t}_3)$ and $(x_3 z, \boldsymbol{t}_3)$, which proves the lemma. Otherwise, Equation (7) implies

$$(x_3 yz, \boldsymbol{t}_3) \in R_q \iff (x_3 z, \boldsymbol{t}_3) \in R_q,$$
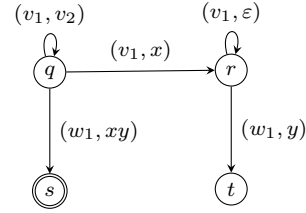

Fig. 2. The pattern witnessing nonrecognizability for deterministic 2-tape automata. Here, either state s is final and t is nonfinal, or vice versa.

which contradicts Equations (5) and (6). Hence, we can set $\boldsymbol{v} = (x_2, \boldsymbol{t}_2)$ and either set $\boldsymbol{w} = (x_3 yz, \boldsymbol{t}_3)$ in case (i) or set $\boldsymbol{w} = (x_3 z, \boldsymbol{t}_3)$ in case (ii). This concludes the proof.

(3 ⇒ 4): Let $\boldsymbol{v}, \boldsymbol{w}$ and $q \in Q$ such that $q \xrightarrow{\boldsymbol{v}} q$, v_1 is null-transparent, R_q separates $\boldsymbol{w}$ and $(v_1, \boldsymbol{\varepsilon})\boldsymbol{w}$. Let $m > |w_2 \cdots w_k| + 1$. Let ρ be the run on $(v_1, \boldsymbol{\varepsilon})^m \boldsymbol{w}$ starting in q. It contains a subrun reading $(v_1, \boldsymbol{\varepsilon})$ between two Q_1-states, i.e. we can factor $(w_2, \ldots, w_k) = \boldsymbol{xy}$ such that

$$\rho: q \xrightarrow{(v_1^{i-1}, \boldsymbol{x})} s \xrightarrow{(v_1, \boldsymbol{\varepsilon})} r \xrightarrow{(v_1^{m-i} w_1, \boldsymbol{y})} t$$

for some $s, r \in Q_1$. Since v_1 is null-transparent there is a cycle $r \xrightarrow{(v_1, \boldsymbol{\varepsilon})} r$. Therefore $V \stackrel{\text{def}}{=} \{(v_1, \boldsymbol{\varepsilon})^j \boldsymbol{w} \mid j \geq m\}$ is either contained in R_q or disjoint from R_q. Since R_q separates $\boldsymbol{w}$ and $(v_1, \boldsymbol{\varepsilon})\boldsymbol{w}$, it also separates one of them from V. If R_q separates $\boldsymbol{w}$ and V, then $\boldsymbol{v}^m$ and $\boldsymbol{w}$ satisfy the condition from the proposition. Otherwise, R_q separates $(v_1, \boldsymbol{\varepsilon})\boldsymbol{w}$ and V, and the tuples $\boldsymbol{v}^m$ and $(v_1, \boldsymbol{\varepsilon})\boldsymbol{w}$ satisfy the condition from the proposition. □

B. Polynomial-time algorithm for binary relations

From Proposition 2 we can derive a pattern which is present in $\mathcal{A}$ if and only if R is *not* recognizable. For binary relations the pattern is visualized in Figure 2. This pattern can be detected in polynomial-time by reducing to the inequivalence problem for binary deterministic rational relations.

Proposition 3. $\approx_1$ *has infinite index if and only if there exist words* $v_1, w_1 \in \Sigma^*$, *tuples* $\boldsymbol{v}_2, \boldsymbol{x}, \boldsymbol{y} \in (\Sigma^*)^{k-1}$, *and states* $q, r \in Q$ *such that*

1) $q \xrightarrow{(v_1, \boldsymbol{v}_2)} q$, $q \xrightarrow{(v_1, \boldsymbol{x})} r$, $r \xrightarrow{(v_1, \boldsymbol{\varepsilon})} r$,
2) $(w_1, \boldsymbol{xy}) \in R_q \iff (w_1, \boldsymbol{y}) \notin R_r$.

Proof. For the "if" direction observe that R_q separates $(w_1, \boldsymbol{xy})$ and $(v_1, \boldsymbol{\varepsilon})^+(w_1, \boldsymbol{xy})$. Therefore $\approx_1$ has infinite index by Proposition 2 point 4). For the "only if" direction assume that $\approx_1$ has infinite index. Again, by Proposition 2 point 4) there exist $(v_1, \boldsymbol{v}_2), (w_1, \boldsymbol{w}_2) \in (\Sigma^*)^k$ and a state $q \in Q$ such that $q \xrightarrow{(v_1, \boldsymbol{v}_2)} q$, and R_q separates $(w_1, \boldsymbol{w}_2)$ and $(v_1, \boldsymbol{\varepsilon})^+(w_1, \boldsymbol{w}_2)$. Let $m > \|\boldsymbol{w}_2\| + 1$ and let ℓ be such that v_1^ℓ is null-transparent. Consider the unique run ρ_q on $(v_1, \boldsymbol{\varepsilon})^{m\ell}(w_1, \boldsymbol{w}_2)$ starting from q. It must contain a subrun of the form $s \xrightarrow{(v_1, \boldsymbol{\varepsilon})^\ell} r$ where $s, r \in Q_1$. Hence we can factorize $\boldsymbol{w}_2 = \boldsymbol{xy}$ such that ρ_q has the form

$$\rho_q: q \xrightarrow{(v_1^{(i-1)\ell}, \boldsymbol{x})} s \xrightarrow{(v_1, \boldsymbol{\varepsilon})^\ell} r \xrightarrow{(v_1^{(m-i)\ell} w_1, \boldsymbol{y})} t. \quad (8)$$

Since v_1^ℓ is null-transparent there exists a cycle $r \xrightarrow{(v_1,\boldsymbol{\varepsilon})^\ell} r$. Since $\mathcal{A}$ is deterministic, this allows us to choose $i = m$ in Equation (8) and write

$$\rho_q\colon \; q \xrightarrow{(v_1^{(m-1)\ell},\boldsymbol{x})} s \xrightarrow{(v_1,\boldsymbol{\varepsilon})^\ell} r \xrightarrow{(w_1,\boldsymbol{y})} t. \qquad (9)$$

Since R_q separates $(w_1, \boldsymbol{w}_2) = (w_1, \boldsymbol{xy})$ and $(v_1, \boldsymbol{\varepsilon})^{m\ell}(w_1, \boldsymbol{w}_2)$, we know that $(w_1, \boldsymbol{xy}) \in R_q$ if and only if $(w_1, \boldsymbol{y}) \notin R_r$. Hence the words $v_1^{m\ell}, w_1$ together with the tuples $\boldsymbol{v}_2^{m\ell}, \boldsymbol{x}, \boldsymbol{y}$ satisfy the claim. $\qquad \square$

Theorem 6. *The recognizability problem for binary deterministic rational relations is logspace reducible to the equivalence problem for binary deterministic rational relations.*

Proof. Let R be a binary deterministic rational relation, which is recognizable if and only if $\approx_1$ has finite index by Proposition 1. By Proposition 3 this holds if and only if for all state pairs $q, r \in Q$ and all words $v_1, w_1, v_2, x, y \in \Sigma^*$ the following two conditions are equivalent:

(C1) $q \xrightarrow{(v_1,v_2)} q$, $q \xrightarrow{(v_1,x)} r$, $r \xrightarrow{(v_1,\varepsilon)} r$, $(w_1, xy) \in R_q$
(C2) $q \xrightarrow{(v_1,v_2)} q$, $q \xrightarrow{(v_1,x)} r$, $r \xrightarrow{(v_1,\varepsilon)} r$, $(w_1, \boldsymbol{y}) \in R_r$

Using an appropriate encoding we can reduce the equivalence of (C1) and (C2) to the equivalence problem for binary deterministic rational relations.

Suppose π, ρ are runs which read the same input word (in our case, this would be v_1), i.e. we can write

$$\pi\colon \; s_1 \xrightarrow{g_0} t_1 \xrightarrow{a_1} s_2 \xrightarrow{g_1} t_2 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n \xrightarrow{g_n} t_n$$

and

$$\rho\colon \; s_1' \xrightarrow{h_0} t_1' \xrightarrow{a_1} s_2' \xrightarrow{h_1} t_2' \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n' \xrightarrow{h_n} t_n'$$

where each a_i is a letter and the states t_i, t_i' are precisely the states in π and ρ in Q_1. We define their synchronized shuffle $\pi \sqcup\!\sqcup\, \rho \in (\Sigma \cup \{\diamond\})^*$ as

$$\pi \sqcup\!\sqcup\, \rho = g_0 \diamond h_0 \diamond a_1 \diamond g_1 \diamond h_1 \diamond a_2 \diamond \cdots \diamond a_n \diamond g_n \diamond h_n.$$

We encode (C1) as the binary relation

$$C_1 = \{(q\,r\,w_1, (\pi \sqcup\!\sqcup\, \rho)\,\$\,y) \mid q, r \in Q, \; \pi\colon q \xrightarrow{(v_1,v_2)} q,$$
$$\rho\colon q \xrightarrow{(v_1,x)} r, \; r \xrightarrow{(v_1,\varepsilon)} r, \; (w_1, xy) \in R_q\}$$

and (C2) as the binary relation

$$C_2 = \{(q\,r\,w_1, (\pi \sqcup\!\sqcup\, \rho)\,\$\,y) \mid q, r \in Q, \; \pi\colon q \xrightarrow{(v_1,v_2)} q,$$
$$\rho\colon q \xrightarrow{(v_1,x)} r, \; r \xrightarrow{(v_1,\varepsilon)} r, \; (w_1, y) \in R_r\}.$$

Observe that $C_1 = C_2$ if and only if (C1) and (C2) are equivalent. It remains to verify that C_1 and C_2 are deterministic rational and we can construct automata in logspace. First, for each state pair $q, r \in Q$ we can construct a DFA over $\Sigma \cup \{\diamond\}$ which accepts precisely the synchronized shuffles $\pi \sqcup\!\sqcup\, \rho$ where $\pi\colon q \xrightarrow{(v_1,v_2)} q$, $\rho\colon q \xrightarrow{(v_1,x)} r$ and $r \xrightarrow{(v_1,\varepsilon)} r$ for some words v_1, v_2, x. Since x can be easily extracted as a subword of $\pi \sqcup\!\sqcup\, \rho$, a deterministic transducer can verify whether the input pair $(q\,r\,w_1, (\pi \sqcup\!\sqcup\, \rho)\,\$\,y)$ satisfies $(w_1, xy) \in R_q$ and whether it satisfies $(w_1, y) \in R_r$. $\qquad \square$

C. Arbitrary relations

The approach from Theorem 6 does not work for arity $k \geq 3$. The issue is that the words v_2, x, y from (C1) and (C2) would become $(k-1)$-tuples $\boldsymbol{v}_2, \boldsymbol{x}, \boldsymbol{y}$. It is not clear how to appropriately encode the runs on $(v_1, \boldsymbol{x})$ and $(w_1, \boldsymbol{xy})$ in (C1) so that they can be simulated by an automaton. Still, we can express (the negation of) property 3) in Proposition 2 as the equivalence of two polynomial space constructible deterministic multitape automata. Since equivalence of deterministic k-tape automata is in coNP [33], and in coRP for fixed k [8], the complexity bounds from Theorem 3 follow.

Theorem 7. *The recognizability problem for k-ary deterministic rational relations is polynomial space reducible to the equivalence problem for k-ary deterministic rational relations.*

Proof. Let R be a k-ary deterministic rational relation, which is recognizable if and only if $\approx_j$ has finite index for all $j < k$ by Proposition 1. It suffices to show how to reduce the test whether $\approx_1$ has finite index to the equivalence problem of polynomial space constructible deterministic k-tape automata. By Proposition 2 point 3), $\approx_1$ has finite index if and only if for all states $q \in Q$ and all tuples $\boldsymbol{v}, \boldsymbol{w} \in (\Sigma^*)^k$ the following two conditions are equivalent:

(P1) $q \xrightarrow{\boldsymbol{v}} q$, v_1 null-transparent, $\boldsymbol{w} \in R_q$
(P2) $q \xrightarrow{\boldsymbol{v}} q$, v_1 null-transparent, $(v_1, \boldsymbol{\varepsilon})\boldsymbol{w} \in R_q$

We encode (P1) and (P2) as deterministic rational relations. First observe that we can construct an exponentially large DFA for the language of all null-transparent words $v_1 \in \Sigma^+$. It simulates runs on v_1 in parallel from every state $s \in Q_1$, and verifies that $s \xrightarrow{(v_1,\varepsilon)} t$ implies $t \xrightarrow{(v_1,\varepsilon)} t$. We encode a run π as an alternating sequence $\mathsf{flat}(\pi) \in (Q\Sigma)^*Q$ of states and input letters. Under this encoding, valid runs can be recognized by a polynomially sized DFA. Define the following k-ary relations

$$P_1 = \{(q\,\mathsf{flat}(\pi)\,\$, \boldsymbol{\varepsilon})\,\boldsymbol{w} \mid q \in Q, \; v_1 \text{ null-transparent},$$
$$\pi\colon q \xrightarrow{\boldsymbol{v}} q, \; \boldsymbol{w} \in R_q\}$$

and

$$P_2 = \{(q\,\mathsf{flat}(\pi)\,\$, \boldsymbol{\varepsilon})\,\boldsymbol{w} \mid q \in Q, \; v_1 \text{ null-transparent},$$
$$\pi\colon q \xrightarrow{\boldsymbol{v}} q, \; (v_1, \boldsymbol{\varepsilon})\boldsymbol{w} \in R_q\}.$$

Since v_1 can be easily extracted from $\mathsf{flat}(\pi)$, we can construct exponentially sized deterministic k-tape automata for P_1 and P_2. Furthermore, the conditions (P1) and (P2) are equivalent if and only if $P_1 = P_2$. $\qquad \square$

D. Reducing equivalence to recognizability

Let us complement the presented algorithms for recognizability with the following "converse direction". We solved the recognizability problem by reducing to the equivalence problem over deterministic rational relations (in logspace, for binary relations). In fact, the equivalence problem is logspace reducible to the recognizability problem (for arbitrary arity).

Theorem 8. *Let $k \geq 2$. The equivalence problem for k-ary deterministic rational relations is logspace reducible to the recognizability problem for k-ary deterministic rational relations.*

Proof. Given two deterministic k-tape automata $\mathcal{A}$ and $\mathcal{B}$. First we ensure that both $R(\mathcal{A})$ and $R(\mathcal{B})$ are finite relations, and, in particular, recognizable. By [33] the automata $\mathcal{A}$ and $\mathcal{B}$ are equivalent if and only if they accept the same tuples of length at most $n-1$, where n is the total number of states in $\mathcal{A}$ and $\mathcal{B}$. We can compute in logspace a deterministic k-tape automaton $\mathcal{A}'$ such that $R(\mathcal{A}') = R(\mathcal{A}) \cap \{\boldsymbol{u} \in (\Sigma^*)^k \mid \|\boldsymbol{u}\| < n\}$, and analogously $\mathcal{B}'$ for $\mathcal{B}$. The automaton $\mathcal{A}'$ tracks the length of the prefix tuple read so far, up to threshold n, and rejects all tuples of length at least n.

We claim that $R(\mathcal{A}') = R(\mathcal{B}')$ if and only if

$$
\begin{aligned}
T = \{(a^i\#, a^i\#, \boldsymbol{\varepsilon}) \mid i \in \mathbb{N}\} R(\mathcal{A}')\ \cup \\
\{(a^i\#, a^j\#, \boldsymbol{\varepsilon}) \mid i \neq j\} R(\mathcal{B}')
\end{aligned}
$$

is recognizable where a and $\#$ are fresh distinct letters. Observe that a deterministic k-tape automaton for T is logspace computable from $\mathcal{A}'$ and $\mathcal{B}'$. If $R(\mathcal{A}') = R(\mathcal{B}')$ then

$$
T = \{(a^i\#, a^j\#, \boldsymbol{\varepsilon}) \mid i, j \in \mathbb{N}\} R(\mathcal{A}')
$$

is the concatenation of two recognizable relations and hence itself recognizable. Suppose that $R(\mathcal{A}') \neq R(\mathcal{B}')$ and assume that there exists a tuple $\boldsymbol{v} \in R(\mathcal{A}') \setminus R(\mathcal{B}')$ (the case where $R(\mathcal{B}') \setminus R(\mathcal{A}') \neq \emptyset$ is similar). If T would be recognizable then $T\boldsymbol{v}^{-1} = \{\boldsymbol{u} \mid \boldsymbol{u}\boldsymbol{v} \in T\}$ would also be recognizable. However

$$
T\boldsymbol{v}^{-1} = \{(a^i\#, a^i\#, \boldsymbol{\varepsilon}) \mid i \in \mathbb{N}\}
$$

is clearly not recognizable. $\qquad\square$

E. Constructing an independent automaton

Theorem 2 raises the question how to translate a deterministic multitape automaton into an equivalent automaton with independent tapes, if one exists. Such a construction will be needed in the next section, to decide whether a deterministic multitape automaton recognizes a synchronous relation. Formally, an *independent k-tape automaton* $\mathcal{I}$ is a tuple $\mathcal{I} = (\mathcal{A}_1, \ldots, \mathcal{A}_k, F)$ consisting of DFAs $\mathcal{A}_i$ without final states and a set of state tuples $F \subseteq Q_1 \times \cdots \times Q_k$, where Q_i is the state set of $\mathcal{A}_i$. The relation $R(\mathcal{I})$ recognized by $\mathcal{I}$ is the set of all tuples $(w_1, \ldots, w_k)$ such that for each $i \in [1, k]$ the unique run of $\mathcal{A}_i$ on w_i ends in a state $q_i \in Q_i$ with $(q_1, \ldots, q_k) \in F$. Note that independent multitape automata recognize exactly the relations in **Rec**.

Theorem 9. *Given a deterministic k-tape automaton for a recognizable relation R, one can compute an independent k-tape automaton for R in double exponential time.*

Proof. For each $j \in [1, k]$ define the relation $\equiv_j$ by

$$
x \equiv_j y \iff^{\text{def}} \text{ for all } z \in \Sigma^*: xz \approx_j yz,
$$

which is a right-congruence, i.e. $x \equiv_j y$ implies $xa \equiv_j ya$. Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ be a deterministic k-tape automaton

for a recognizable relation R. Suppose that x, y, z are words where y is $nn!$-invisible in the context of x. Then $xyz \equiv_j xz$ since otherwise $xy(zz') \not\approx_j x(zz')$ for some word z', which contradicts Proposition 2. Hence, for each word w of length $f(nn!) + 1$ there exists an $\equiv_j$-equivalent word w' of length at most $f(nn!)$, by cutting out $nn!$-invisible factor according to Lemma 11 where $f(N) \overset{\text{def}}{=} 2(Nn)^n$. Furthermore, the function $w \mapsto w'$ can be computed in double exponential time, as remarked in [15, Section 8].

Hence, the independent k-tape automaton $(\mathcal{A}_1, \ldots, \mathcal{A}_k, F)$ works as follows. The states of $\mathcal{A}_j$ are words of length at most $f(nn!)$. The initial state is the empty word ε. If the current state (word) is w and the next input symbol is $a \in \Sigma$, then the next state is the word obtained from wa by removing an $nn!$-invisible factor, if possible. In this way, at each time step the reached state is a word that is $\equiv_j$-equivalent to the read prefix. Finally, a tuple $\boldsymbol{v}$ is marked final if and only if $\boldsymbol{v} \in R$. On input tuple $(w_1, \ldots, w_k)$ each DFA $\mathcal{A}_j$ reaches a state v_j with $v_j \equiv_j w_j$, and therefore $(v_1, \ldots, v_k) \in R$ if and only if $(w_1, \ldots, w_k) \in R$. $\qquad\square$

We remark that the double exponential bound in Theorem 9 is optimal, which can be derived from the proof by Meyer and Fischer for the double exponential succinctness gap between DPDAs and DFAs [45]. To keep the paper self-contained, we provide an alternative proof.

Proposition 4. *There exists a recognizable relation $R_n \subseteq \{0, 1\}^* \times \{0, 1\}^*$ which is accepted by a deterministic 2-tape automaton with $O(n^2 \log n)$ states so that any independent 2-tape automaton for R_n has in total at least $2^{2^{n-1}}$ states.*

Proof. Let $R_n \subseteq [1, n]^* \times [1, n]^*$ be the relation containing all pairs (u, v) where $|v| \leq 2n$ and v is a scattered subword of u. Observe that R_n is accepted by a deterministic 2-tape automaton with $O(n)$ states. Then $\approx_1^{R_n}$ is *Simon's congruence* with parameter $2n$ [12]. Its index is finite and bounded from below by $2^{2^{n-1}}$ by [46, Theorem 1.2]. If an independent 2-tape automaton $\mathcal{I} = (\mathcal{A}_1, \mathcal{A}_2, F)$ recognizes R_n then the index of $\approx_1^{R_n}$ is a lower bound for the number of states of $\mathcal{A}_1$. Finally, we can replace the alphabet $[1, n]$ by codes from $\{0, 1\}^{\log n}$, increasing the automaton size by a $\log n$-factor. $\qquad\square$

VI. Deciding Synchronicity in **DRat**

In this section we prove Theorem 3 by showing that synchronicity can be reduced to recognizability for relations in **DRat**, which can be solved according to Theorem 2. Let us remark that there also exists a reduction in the reverse direction, whose proof can be found in Appendix B.

Proposition 5. *Given a k-tape automaton $\mathcal{A}$ for a relation R, one can compute in logspace a k-tape automaton $\mathcal{B}$ for a relation S such that R is recognizable if and only if S is synchronous. If $\mathcal{A}$ is deterministic, then so is $\mathcal{B}$.*

In the rest of this section we show Theorem 3. We first give an intuition for the case $k = 2$. Suppose R is given by a deterministic 2-tape automaton $\mathcal{A}$ with the property

Fig. 3. An asynchronous cycle can produce words (u_1, u_2) with unbounded length difference. The words s_i are pairwise inequivalent words with respect to $\approx_1^R$, separated by the words $t_{i,j}$.

Fig. 4. If there are h pairwise $\approx_{[1,2]}^{R_q}$-inequivalent tuples $\boldsymbol{s_1}, \ldots, \boldsymbol{s_h}$ then also the Myhill-Nerode equivalence $\sim_{\otimes R}$ has at least h classes (witnessed by $x_1, \ldots, x_h$).

that every reachable cycle $p \xrightarrow{(v_1, v_2)} p$ satisfies $|v_1| = |v_2|$. This ensures that $\mathcal{A}$ has *bounded delay* [22, Section 3], i.e. the head positions cannot be arbitrarily far apart during the computation of $\mathcal{A}$. In fact, the delay is bounded by the number of states $|Q|$. It is well-known that such an automaton recognizes a synchronous relation [22, Corollary 3.4], since letters that are "read ahead" on a tape can be stored in a queue of length $|Q|$. Let us now consider the case where $\mathcal{A}$ contains an *asynchronous* cycle of the form $p \xrightarrow{(v_1, v_2)} p$ with $|v_1| < |v_2|$ (the case $|v_1| > |v_2|$ is symmetric). We partition the automaton into an asynchronous part, containing all states which are reachable from an asynchronous cycle, and a synchronous part. While the simulation using a queue works in the synchronous part of the automaton, the delay can become unbounded in the asynchronous part, by traversing asynchronous cycles repeatedly.

We claim that R is synchronous if and only if for each state q in the asynchronous part, the relation R_q is recognizable. If each such relation R_q is recognizable and in particular synchronous, then the computation from state q can be continued synchronously using a synchronous automaton for R_q. For the other direction, assume that R_q is not recognizable for some asynchronous state q. Therefore $\approx_1^{R_q}$ has infinite index by Proposition 1, i.e. there exist words $(s_i)_{i \geq 1}$ and $(t_{i,j})_{i<j}$ such that R_q separates $(s_i, t_{i,j})$ and $(s_j, t_{i,j})$ for all $i < j$. We claim that the Myhill-Nerode equivalence relation $\sim_{\otimes R}$ of the language $\otimes R$ of convolutions has at least h classes for each $h \in \mathbb{N}$: Take an asynchronous cycle $p \xrightarrow{(v_1, v_2)} p$ from which q is reachable. We can produce runs $q_0 \xrightarrow{(u_1, u_2)} q$ where the delay $|u_2| - |u_1|$ is arbitrary large. Pick such a run where $|u_2| - |u_1| \geq \max\{|s_1|, \ldots, |s_h|\}$. Then any two words $(u_1 s_i) \otimes u_2$ and $(u_1 s_j) \otimes u_2$ for $1 \leq i < j \leq h$ are inequivalent with respect to $\sim_{\otimes R}$ because $\otimes R$ separates $(u_1 s_i) \otimes (u_2 t_{i,j})$ and $(u_1 s_j) \otimes (u_2 t_{i,j})$, see the illustration in Figure 3. Thus, the synchronicity problem can be reduced to checking recognizability of relations R_q where q is reachable from an asynchronous cycle.

Let us now consider the general case of a k-ary deterministic rational relation R. Again, we can ignore the endmarker $\dashv$ since appending $(\dashv, \ldots, \dashv)$ to R preserves (non)synchronicity. Hence, for the rest of this section we assume that R is given by a deterministic k-tape automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ with $R = R(\mathcal{A})$. Moreover, we assume that every state in Q is reachable from q_0. A cycle in $\mathcal{A}$ reading $(v_1, \ldots, v_k)$ *induces* a partition P on the components $[1, k]$ where two components i and j are in the same block in P if and only if $|v_i| = |v_j|$. For a state $q \in Q$ we define the partition P_q as the coarsest

refinement of all partitions induced by a cycle from which q is reachable. As before, let R_q be the relation recognized from state q.

Lemma 12. *For every $q \in Q$, P_q is computable in time polynomial in the size of $\mathcal{A}$.*

Proof. The algorithm proceeds as follows. As a first step we compute for each $q \in Q$ the coarsest refinement S_q of all partitions that are induced by *simple* cycles on q. Check for every $1 \leq i < j \leq k$ if there exists a simple cycle $q \xrightarrow{v} q$ such that $|v_i| \neq |v_j|$ and if so, store it as a constraint that i and j are in different blocks. Then S_q is the coarsest partition of $[1, k]$ that fulfills all stored constraints. Note that the existence of a simple cycle $q \xrightarrow{v} q$ with $|v_i| \neq |v_j|$ can be checked in nondeterministic logspace by storing the current length difference of the words in the i-th and j-th component on the guessed path in a counter whose value is bounded by $|Q|$.

We claim that $P_q = S_{q_1} \sqcap \cdots \sqcap S_{q_n}$ where $q_1, \ldots, q_n$ are the states from which q is reachable. By definition, P_q is finer than $S_{q_1} \sqcap \cdots \sqcap S_{q_n}$. For the other direction let P be a partition induced by a cycle c from which q is reachable. For the sake of contradiction assume that there exist $1 \leq i < j \leq k$ that are in different blocks in P but in the same block in S_{q_ℓ} for all $\ell \in [1, n]$. Since any cycle contains a simple cycle, there exists a simple cycle $p \xrightarrow{v} p$ that is contained in c. By assumption, it holds that $|v_i| = |v_j|$ which means that after removing $p \xrightarrow{v} p$ from c, the cycle c still induces a partition where i and j are in different blocks. Furthermore, q is still reachable from c. We can repeat this argument until c is a simple cycle inducing a partition where i and j are in different blocks, a contradiction. □

Recall that, for binary relations we tested recognizability for all states reachable from asynchronous cycles. For higher arity relations, we need to test whether each relation R_q conforms to P_q.

Lemma 13. *If R_q does not conform to P_q for some state $q \in Q$, then R is not synchronous.*

Proof. Assume that R_q does not conform to P_q. By Theorem 4 there exists a partition P induced by a cycle $p \xrightarrow{v} p$ so that q is reachable from p and R_q does not conform to P. By permuting components, we can assume that $|v_1| \leq \cdots \leq |v_k|$. Hence P is a partition of $[1, k]$ into intervals $B_1, \ldots, B_n$, which are listed in ascending order. Since the intervals $B_1 \cup \cdots \cup B_i$ for $i \in [1, n]$ generate P, there exists an index $r \in [1, n]$ such

that $\approx^{R_q}_{B_1 \cup \cdots \cup B_r}$ has infinite index by Lemma 2. Set $[1, m] \stackrel{\text{def}}{=} B_1 \cup \cdots \cup B_r$. Observe that for any number $b \in \mathbb{N}$ there exists a run $q_0 \xrightarrow{\boldsymbol{u}} q$ such that $|u_i| + b \leq |u_j|$ for all $i \in [1, m]$ and $j \in [m+1, k]$. Such runs can be constructed by traversing the cycle $p \xrightarrow{\boldsymbol{v}} p$ sufficiently often.

We show that for every $h \in \mathbb{N}$, the Myhill-Nerode equivalence relation $\sim_{\otimes R}$ of the language $\otimes R$ of convolutions has at least h classes. This proves that $\otimes R$ is not regular and therefore R is not synchronous.

Let $h \in \mathbb{N}$. Since $\approx^{R_q}_{[1,m]}$ has infinite index there are tuples $\boldsymbol{s_i} \stackrel{\text{def}}{=} (s_{i,1}, \ldots, s_{i,m})$ for $i \in [1, h]$ and $\boldsymbol{t_{i,j}} \stackrel{\text{def}}{=} (t_{i,j,m+1}, \ldots, t_{i,j,k})$ for $1 \leq i < j \leq h$ such that $(\boldsymbol{s_i}, \boldsymbol{t_{i,j}}) \in R_q$ if and only if $(\boldsymbol{s_j}, \boldsymbol{t_{i,j}}) \notin R_q$ for all $1 \leq i < j \leq h$.

Let $b \stackrel{\text{def}}{=} \max\{|s_{i,j}| \mid i \in [1, h], j \in [1, m]\}$. By the observation above there exists a run $q_0 \xrightarrow{\boldsymbol{u}} q$ such that $|u_i| + b \leq |u_j|$ for all $i \in [1, m]$ and $j \in [m+1, k]$. Therefore, there exists a number ℓ such that all words in the m-tuple $(u_1, \ldots, u_m)\boldsymbol{s_i}$ have length at most ℓ, and all words in the $(k-m)$-tuple $(u_{m+1}, \ldots, u_k)$ have length at least ℓ, see Figure 4 for an illustration. Since $\mathcal{A}$ is deterministic we have $\boldsymbol{u}(\boldsymbol{s_i}, \boldsymbol{t_{i,j}}) \in R$ if and only if $\boldsymbol{u}(\boldsymbol{s_j}, \boldsymbol{t_{i,j}}) \notin R$ for all $1 \leq i < j \leq h$. This can be turned into a proof that $\otimes R$ has at least h Myhill-Nerode classes. For a word w of length at least ℓ, we denote by $\mathrm{pre}_\ell(w)$ the prefix of w of length ℓ and by $\mathrm{suf}_\ell(w)$ the suffix of w after $\mathrm{pre}_\ell(w)$. For all $i \in [1, h]$ define

$$x_i \stackrel{\text{def}}{=} u_1 s_{i,1} \otimes \cdots \otimes u_m s_{i,m} \otimes \mathrm{pre}_\ell(u_{m+1}) \otimes \cdots \otimes \mathrm{pre}_\ell(u_k)$$

and for all $1 \leq i < j \leq h$ define

$$y_{i,j} \stackrel{\text{def}}{=} \varepsilon \otimes \cdots \otimes \varepsilon \\ \otimes \mathrm{suf}_\ell(u_{m+1} t_{i,j,m+1}) \otimes \cdots \otimes \mathrm{suf}_\ell(u_k t_{i,j,k}).$$

Observe that $x_i y_{i,j}$ and $x_j y_{i,j}$ are the convolutions of the tuples $\boldsymbol{u}(\boldsymbol{s_i}, \boldsymbol{t_{i,j}})$ and $\boldsymbol{u}(\boldsymbol{s_j}, \boldsymbol{t_{i,j}})$, respectively, and therefore $x_i \not\sim_{\otimes R} x_j$ for all $1 \leq i < j \leq h$. $\square$

Testing whether a relation conforms to a partition is an, a priori, more difficult problem than recognizability, and it is not clear how to decide it for deterministic rational relations. Instead, we will summarize the components inside each partition block into a single component, and test recognizability for the summarized relation.

In the following we always assume that the blocks of a partition $P = \{B_1, \ldots, B_n\}$ are ordered so that $\min(B_1) < \cdots < \min(B_n)$, and each block $B_i = \{b_{i,1}, \ldots, b_{i,|B_i|}\}$ is given such that $b_{i,1} < \cdots < b_{i,|B_i|}$. For a relation $R \subseteq (\Sigma^*)^k$ and the partition P of $[1, k]$ as above we define the *summarized relation*

$$R^P \stackrel{\text{def}}{=} \{(u_{b_{1,1}} \otimes \cdots \otimes u_{b_{1,|B_1|}}, \ldots, u_{b_{n,1}} \otimes \cdots \otimes u_{b_{n,|B_n|}}) \\ \mid (u_1, \ldots, u_k) \in R\}.$$

We write $R_q^\otimes$ for $R_q^{P_q}$. Under the assumption that $R_q^\otimes$ is deterministic rational, we can test if R_q conforms to P_q.

Lemma 14. *If R^P is deterministic rational, then R^P is recognizable if and only if R conforms to P.*

Proof. Let $P = \{B_1, \ldots, B_n\}$. By Proposition 1 the summarized relation R^P is recognizable if and only if $\approx^{R^P}_{[1,i]}$ has finite index for all $i \in [1, n-1]$. Let $B_{[1,i]} \stackrel{\text{def}}{=} B_1 \cup \cdots \cup B_i$. By Lemma 2 we have that R conforms to P if and only if $\approx^R_{B_{[1,i]}}$ has finite index for all $i \in [1, n-1]$. The claim follows since $\approx^{R^P}_{[1,i]}$ has finite index if and only if $\approx^R_{B_{[1,i]}}$ has finite index. $\square$

We are now ready to give the reduction from synchronicity to recognizability proving Theorem 3. For a partition P of $[1, k]$ we write $Q_P \stackrel{\text{def}}{=} \{q \in Q \mid P_q = P\}$. We partition Q into layers $L_1, \ldots, L_k$ where $L_t = \{q \in Q \mid |P_q| = t\}$. Observe that all states reachable from a state $q \in L_t$ are contained in $L_t \cup \cdots \cup L_k$. The algorithm processes the layers $L_k, L_{k-1}, \ldots, L_1$ in descending order. For each state q in layer L_t the algorithm (i) constructs a deterministic multitape automaton $\mathcal{A}_q^\otimes$ for $R_q^\otimes$, (ii) tests whether $R_q^\otimes$ is recognizable, (iii) and, if so, constructs an independent multitape automaton $\mathcal{I}_q^\otimes$ for $R_q^\otimes$. For layer L_k the automaton $\mathcal{A}_q^\otimes$ is simply the automaton $\mathcal{A}$ with initial state q. For the other layers the automaton $\mathcal{A}_q^\otimes$ will be built from the automata $\mathcal{I}_{q'}^\otimes$ from the previous layers, which will be explained below (Lemma 15). The automata $\mathcal{I}_q^\otimes$ are constructed from $\mathcal{A}_q^\otimes$ using Theorem 9, which is correct under the assumption that $R_q^\otimes$ is indeed recognizable. If one of the recognizability tests is negative, the algorithm terminates and reports that R is not synchronous. Otherwise, if all recognizability tests succeed, the algorithm reports that R is synchronous.

Let us argue that the algorithm is correct. If one of the relations $R_q^\otimes$ is deterministic rational but not recognizable, then R is indeed not synchronous by Lemmas 13 and 14. If all recognizability tests succeed, then in particular the relation $R_{q_0}^\otimes$ is recognizable. This easily implies synchronicity of R. In fact, we can make the algorithm slightly more efficient. Observe that the recognizability tests in layer L_1 will always be positive since the relations $R_q^\otimes$ in layer L_1 are regular languages. Therefore, we can skip processing the last layer L_1 and also skip constructing the independent multitape automata in layer L_2.

It remains to show how to construct the automaton $\mathcal{A}_q^\otimes$ witnessing that $R_q^\otimes$ is deterministic rational.

Lemma 15. *Given $q \in L_t$ and independent multitape automata $\mathcal{I}_{q'}^\otimes$ for $R_{q'}^\otimes$ for all $q' \in L_{t+1} \cup \cdots \cup L_k$, one can compute $\mathcal{A}_q^\otimes$ in time polynomial in the sizes of the $\mathcal{I}_{q'}^\otimes$ and exponential in the size of $\mathcal{A}$.*

The detailed construction can be found in Appendix B. The idea is that $\mathcal{A}_q^\otimes$ consists of two parts. Let $q \in Q_P$ for a partition $P = \{B_1, \ldots, B_n\}$. In the first part $\mathcal{A}_q^\otimes$ simulates $\mathcal{A}$ over the states in Q_P. In that part it is possible for $\mathcal{A}_q^\otimes$ to read the components within a class of P synchronously since by definition of Q_P the difference of the tape positions between those components is bounded by $|Q|$. Thus, it suffices for $\mathcal{A}_q^\otimes$ to store a word (read like a queue) of length at most

$|Q|$ for each component and simulate $\mathcal{A}$ either on the next symbol in the queue or on the current input symbol if the queue is empty. If it simulates $\mathcal{A}$ on the input symbol of the corresponding tape, we store the symbols read in the other components in the queues of that components. The simulation of $\mathcal{A}$ on the queue is handled like an ε-transition where nothing is read from the input. Those ε-transitions can be removed and do not lead to nondeterminism since there is no branching of ε-transitions possible. If the simulation of $\mathcal{A}$ leads to a state q' that is not contained in Q_P, i.e., $q' \in Q_{P'}$ for some partition $P' = \{B'_1, \dots, B'_{n'}\}$ that is strictly finer than P, then $\mathcal{A}_q^{\otimes}$ changes to the second part. In the second part $\mathcal{A}_q^{\otimes}$ simulates the independent n'-tape automaton $\mathcal{I}_{q'}^{\otimes} = (\mathcal{A}_1, \dots, \mathcal{A}_{n'}, F_{q'})$. For each class B_i, for $i = 1, \dots, n$, the DFAs $\mathcal{A}_j$ that are responsible for components contained in B_i are simulated in parallel either on the queue or the current input symbol. After the whole input was read, $\mathcal{A}_q^{\otimes}$ checks with final states whether the remaining content of the queues leads in each $\mathcal{A}_i$ to some state f_i such that $(f_1, \dots, f_{n'}) \in F_{q'}$.

Finally, we argue that the running time of the algorithm is $2(k-2)$-fold exponential in the automaton size $|\mathcal{A}|$. Inductively, we prove that in layer L_t we can construct the automata $\mathcal{A}_q^{\otimes}$ in $2(k-t)$-fold exponential time and the automata $\mathcal{I}_q^{\otimes}$ in $2(k-t+1)$-fold exponential time. In particular, the automata sizes are bounded by their respective construction times. In layer L_k the automata $\mathcal{A}_q^{\otimes}$ are constructed in polynomial time. In the other layers L_t the automata $\mathcal{A}_q^{\otimes}$ are constructed by Lemma 15 in $2(k-t)$-fold exponential time. Each automaton $\mathcal{I}_q^{\otimes}$ is constructed in double exponential time in the size of $\mathcal{A}_q^{\otimes}$ (Theorem 9), which is $2(k-t+1)$-fold exponential in $|\mathcal{A}|$. Furthermore, each recognizability test on $\mathcal{A}_q^{\otimes}$ in layer L_t where $t \in [3,k]$ takes double exponential time in $|\mathcal{A}_q^{\otimes}|$ by Theorem 2, which is $2(k-t+1)$-fold exponential in $|\mathcal{A}|$. The relations $R_q^{\otimes}$ in layer L_2 are binary and therefore recognizability can be tested in polynomial time in $|\mathcal{A}_q^{\otimes}|$, which is $2(k-2)$-fold exponential in $|\mathcal{A}|$.

REFERENCES

[1] R. Alur and P. Cerný, "Streaming transducers for algorithmic verification of single-pass list-processing programs," in *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2011, Austin, TX, USA, January 26-28, 2011*, T. Ball and M. Sagiv, Eds. ACM, 2011, pp. 599–610. [Online]. Available: https://doi.org/10.1145/1926385.1926454

[2] T. Chen, M. Hague, A. W. Lin, P. Rümmer, and Z. Wu, "Decision procedures for path feasibility of string-manipulating programs with complex operations," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 49:1–49:30, 2019. [Online]. Available: https://doi.org/10.1145/3290362

[3] W. Thomas, "Church's problem and a tour through automata theory," in *Pillars of Computer Science, Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*, ser. Lecture Notes in Computer Science, A. Avron, N. Dershowitz, and A. Rabinovich, Eds., vol. 4800. Springer, 2008, pp. 635–655. [Online]. Available: https://doi.org/10.1007/978-3-540-78127-1_35

[4] E. Filiot, I. Jecker, C. Löding, and S. Winter, "On equivalence and uniformisation problems for finite transducers," in *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, ser. LIPIcs, I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, Eds., vol. 55. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 125:1–125:14. [Online]. Available: https://doi.org/10.4230/LIPIcs.ICALP.2016.125

[5] P. Barceló, D. Figueira, and L. Libkin, "Graph logics with rational relations," *Log. Methods Comput. Sci.*, vol. 9, no. 3, 2013. [Online]. Available: https://doi.org/10.2168/LMCS-9(3:1)2013

[6] M. O. Rabin and D. S. Scott, "Finite automata and their decision problems," *IBM J. Res. Dev.*, vol. 3, no. 2, pp. 114–125, 1959. [Online]. Available: https://doi.org/10.1147/rd.32.0114

[7] A. Muscholl and G. Puppis, "The many facets of string transducers (invited talk)," in *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, ser. LIPIcs, R. Niedermeier and C. Paul, Eds., vol. 126. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 2:1–2:21. [Online]. Available: https://doi.org/10.4230/LIPIcs.STACS.2019.2

[8] J. Worrell, "Revisiting the equivalence problem for finite multitape automata," in *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, ser. Lecture Notes in Computer Science, F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska, and D. Peleg, Eds., vol. 7966. Springer, 2013, pp. 422–433. [Online]. Available: https://doi.org/10.1007/978-3-642-39212-2_38

[9] M. P. Schützenberger, "On finite monoids having only trivial subgroups," *Inf. Control.*, vol. 8, no. 2, pp. 190–194, 1965. [Online]. Available: https://doi.org/10.1016/S0019-9958(65)90108-7

[10] Y. Zalcstein, "Locally testable languages," *J. Comput. Syst. Sci.*, vol. 6, no. 2, pp. 151–167, 1972. [Online]. Available: https://doi.org/10.1016/S0022-0000(72)80020-5

[11] J. A. Brzozowski and I. Simon, "Characterizations of locally testable events," *Discret. Math.*, vol. 4, no. 3, pp. 243–271, 1973. [Online]. Available: https://doi.org/10.1016/S0012-365X(73)80005-6

[12] I. Simon, "Piecewise testable events," in *Automata Theory and Formal Languages, 2nd GI Conference, Kaiserslautern, May 20-23, 1975*, ser. Lecture Notes in Computer Science, H. Barkhage, Ed., vol. 33. Springer, 1975, pp. 214–222. [Online]. Available: https://doi.org/10.1007/3-540-07407-4_23

[13] T. Place, "Deciding classes of regular languages: The covering approach," in *Language and Automata Theory and Applications - 14th International Conference, LATA 2020, Milan, Italy, March 4-6, 2020, Proceedings*, ser. Lecture Notes in Computer Science, A. Leporati, C. Martín-Vide, D. Shapira, and C. Zandron, Eds., vol. 12038. Springer, 2020, pp. 89–112. [Online]. Available: https://doi.org/10.1007/978-3-030-40608-0_6

[14] S. A. Greibach, "A note on undecidable properties of formal languages," *Math. Syst. Theory*, vol. 2, no. 1, pp. 1–6, 1968. [Online]. Available: https://doi.org/10.1007/BF01691341

[15] L. G. Valiant, "Regularity and related problems for deterministic pushdown automata," *J. ACM*, vol. 22, no. 1, pp. 1–10, 1975. [Online]. Available: https://doi.org/10.1145/321864.321865

[16] R. Valk and G. Vidal-Naquet, "Petri nets and regular languages," *J. Comput. Syst. Sci.*, vol. 23, no. 3, pp. 299–325, 1981. [Online]. Available: https://doi.org/10.1016/0022-0000(81)90067-2

[17] S. Cho and D. T. Huynh, "Finite-automaton aperiodicity is pspace-complete," *Theor. Comput. Sci.*, vol. 88, no. 1, pp. 99–116, 1991. [Online]. Available: https://doi.org/10.1016/0304-3975(91)90075-D

[18] R. E. Stearns, "A regularity test for pushdown machines," *Inf. Control.*, vol. 11, no. 3, pp. 323–340, 1967. [Online]. Available: https://doi.org/10.1016/S0019-9958(67)90591-8

[19] O. Carton, C. Choffrut, and S. Grigorieff, "Decision problems among the main subfamilies of rational relations," *RAIRO Theor. Informatics Appl.*, vol. 40, no. 2, pp. 255–275, 2006. [Online]. Available: https://doi.org/10.1051/ita:2006005

[20] C. C. Elgot and J. E. Mezei, "On relations defined by generalized finite

automata," *IBM J. Res. Dev.*, vol. 9, no. 1, pp. 47–68, 1965. [Online]. Available: https://doi.org/10.1147/rd.91.0047

[21] P. C. Fischer and A. L. Rosenberg, "Multitape one-way nonwriting automata," *J. Comput. Syst. Sci.*, vol. 2, no. 1, pp. 88–101, 1968. [Online]. Available: https://doi.org/10.1016/S0022-0000(68)80006-6

[22] C. Frougny and J. Sakarovitch, "Synchronized rational relations of finite and infinite words," *Theor. Comput. Sci.*, vol. 108, no. 1, pp. 45–82, 1993. [Online]. Available: https://doi.org/10.1016/0304-3975(93)90230-Q

[23] B. Khoussainov and A. Nerode, "Automatic presentations of structures," in *Logical and Computational Complexity. Selected Papers. Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, Indiana, USA, 13-16 October 1994*, ser. Lecture Notes in Computer Science, D. Leivant, Ed., vol. 960. Springer, 1994, pp. 367–392. [Online]. Available: https://doi.org/10.1007/3-540-60178-3_93

[24] A. Blumensath and E. Grädel, "Automatic structures," in *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*. IEEE Computer Society, 2000, pp. 51–62. [Online]. Available: https://doi.org/10.1109/LICS.2000.855755

[25] P. A. Abdulla, B. Jonsson, P. Mahata, and J. d'Orso, "Regular tree model checking," in *Computer Aided Verification, 14th International Conference, CAV 2002,Copenhagen, Denmark, July 27-31, 2002, Proceedings*, ser. Lecture Notes in Computer Science, E. Brinksma and K. G. Larsen, Eds., vol. 2404. Springer, 2002, pp. 555–568. [Online]. Available: https://doi.org/10.1007/3-540-45657-0_47

[26] P. A. Abdulla, B. Jonsson, M. Nilsson, and M. Saksena, "A survey of regular model checking," in *CONCUR 2004 - Concurrency Theory, 15th International Conference, London, UK, August 31 - September 3, 2004, Proceedings*, ser. Lecture Notes in Computer Science, P. Gardner and N. Yoshida, Eds., vol. 3170. Springer, 2004, pp. 35–48. [Online]. Available: https://doi.org/10.1007/978-3-540-28644-8_3

[27] J. Berstel, *Transductions and context-free languages*, ser. Teubner Studienbücher : Informatik. Teubner, 1979, vol. 38. [Online]. Available: https://www.worldcat.org/oclc/06364613

[28] L. P. Lisovik, "The identity problem for regular events over the direct product of free and cyclic semigroups," *Dok. Akad. Nauk USSR*, vol. 6, pp. 410–413, 1979.

[29] C. Löding and C. Spinrath, "Decision problems for subclasses of rational relations over finite and infinite words," *Discret. Math. Theor. Comput. Sci.*, vol. 21, no. 3, 2019. [Online]. Available: http://dmtcs.episciences.org/5141

[30] P. Barceló, C. Hong, X. B. Le, A. W. Lin, and R. Niskanen, "Monadic decomposability of regular relations," in *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, ser. LIPIcs, C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, Eds., vol. 132. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 103:1–103:14. [Online]. Available: https://doi.org/10.4230/LIPIcs.ICALP.2019.103

[31] P. Bergsträßer, M. Ganardi, A. W. Lin, and G. Zetzsche, "Ramsey quantifiers over automatic structures: Complexity and applications to verification," in *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, C. Baier and D. Fisman, Eds. ACM, 2022, pp. 28:1–28:14. [Online]. Available: https://doi.org/10.1145/3531130.3533346

[32] D. Kuske, "Is ramsey's theorem omega-automatic?" in *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, ser. LIPIcs, J. Marion and T. Schwentick, Eds., vol. 5. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010, pp. 537–548. [Online]. Available: https://doi.org/10.4230/LIPIcs.STACS.2010.2483

[33] T. Harju and J. Karhumäki, "The equivalence problem of multitape finite automata," *Theor. Comput. Sci.*, vol. 78, no. 2, pp. 347–355, 1991. [Online]. Available: https://doi.org/10.1016/0304-3975(91)90356-7

[34] E. P. Friedman and S. A. Greibach, "A polynomial time algorithm for deciding the equivalence problem for 2-tape deterministic finite state acceptors," *SIAM J. Comput.*, vol. 11, no. 1, pp. 166–183, 1982. [Online]. Available: https://doi.org/10.1137/0211013

[35] ——, "On equivalence and subclass containment problems for deterministic context-free languages," *Inf. Process. Lett.*, vol. 7, no. 6, pp. 287–290, 1978. [Online]. Available: https://doi.org/10.1016/0020-0190(78)90019-4

[36] E. Grädel, W. Thomas, and T. Wilke, Eds., *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, ser. Lecture Notes in

Computer Science, vol. 2500. Springer, 2002. [Online]. Available: https://doi.org/10.1007/3-540-36387-4

[37] S. S. Cosmadakis, G. M. Kuper, and L. Libkin, "On the orthographic dimension of definable sets," *Inf. Process. Lett.*, vol. 79, no. 3, pp. 141–145, 2001. [Online]. Available: https://doi.org/10.1016/S0020-0190(00)00184-8

[38] V. King, O. Kupferman, and M. Y. Vardi, "On the complexity of parity word automata," in *Foundations of Software Science and Computation Structures, 4th International Conference, FOSSACS 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001, Proceedings*, ser. Lecture Notes in Computer Science, F. Honsell and M. Miculan, Eds., vol. 2030. Springer, 2001, pp. 276–286. [Online]. Available: https://doi.org/10.1007/3-540-45315-6_18

[39] A. P. Sistla, M. Y. Vardi, and P. Wolper, "The complementation problem for büchi automata with appplications to temporal logic," *Theor. Comput. Sci.*, vol. 49, pp. 217–237, 1987. [Online]. Available: https://doi.org/10.1016/0304-3975(87)90008-9

[40] D. Kuske and M. Lohrey, "First-order and counting theories of omega-automatic structures," *J. Symb. Log.*, vol. 73, no. 1, pp. 129–150, 2008. [Online]. Available: https://doi.org/10.2178/jsl/1208358745

[41] J.-E. Pin, "Mathematical foundations of automata theory," https://www.irif.fr/~jep/PDF/MPRI/MPRI.pdf, 2022.

[42] M. Veanes, N. S. Bjørner, L. Nachmanson, and S. Bereg, "Monadic decomposition," *J. ACM*, vol. 64, no. 2, pp. 14:1–14:28, 2017. [Online]. Available: https://doi.org/10.1145/3040488

[43] L. Libkin, "Variable independence for first-order definable constraints," *ACM Trans. Comput. Log.*, vol. 4, no. 4, pp. 431–451, 2003. [Online]. Available: https://doi.org/10.1145/937555.937557

[44] B. Boigelot, L. Bronne, and S. Rassart, "An improved reachability analysis method for strongly linear hybrid systems (extended abstract)," in *Computer Aided Verification, 9th International Conference, CAV '97, Haifa, Israel, June 22-25, 1997, Proceedings*, ser. Lecture Notes in Computer Science, O. Grumberg, Ed., vol. 1254. Springer, 1997, pp. 167–178. [Online]. Available: https://doi.org/10.1007/3-540-63166-6_18

[45] A. R. Meyer and M. J. Fischer, "Economy of description by automata, grammars, and formal systems," in *12th Annual Symposium on Switching and Automata Theory, East Lansing, Michigan, USA, October 13-15, 1971*. IEEE Computer Society, 1971, pp. 188–191. [Online]. Available: https://doi.org/10.1109/SWAT.1971.11

[46] P. Karandikar, M. Kufleitner, and P. Schnoebelen, "On the index of simon's congruence for piecewise testability," *Inf. Process. Lett.*, vol. 115, no. 4, pp. 515–519, 2015. [Online]. Available: https://doi.org/10.1016/j.ipl.2014.11.008

APPENDIX

A. Discussion of Proof of Lemma 3.5 in [19]

At the end of the proof of Lemma 3.5 in [19] we have the following setting: There exists a cycle $q \xrightarrow{(u_2, \boldsymbol{t}_2)} q$ where u_2 is null-transparent, and either $u_2 u_3 vw \not\approx_1^{R_q} u_3 vw$ or $u_2 u_3 w \not\approx_1^{R_q} u_3 w$. Observe this precisely matches the situation in point 3) of Proposition 2. Let us assume that $u_2 u_3 vw \not\approx_1^{R_q} u_3 vw$ holds, i.e. there exists $\boldsymbol{z} \in (\Sigma^*)^{k-1}$ such that R_q separates $(u_2 u_3 vw, \boldsymbol{z})$ and $(u_3 vw, \boldsymbol{z})$. First we have $u_2^{i+1} u_3 vw \not\approx_1^{R_q} u_2^i u_3 vw$ for all $i \geq 0$ since R_q separates $(u_2^{i+1} u_3 vw, \boldsymbol{t}_2^i \boldsymbol{z})$ and $(u_2^i u_3 vw, \boldsymbol{t}_2^i \boldsymbol{z})$. Then, it is claimed that $u_2^{i+j} u_3 vw \not\approx_1^{R_q} u_2^i u_3 vw$ for all $i \geq 0$, $j > 0$. Towards a contradiction assume that $u_2^{i+K} u_3 vw \approx_1^{R_q} u_2^i u_3 vw$ for some $i \geq 0$, $K > 0$. Then, it is deduced that $u_2^{i+\lambda K} u_3 vw \approx_1^{R_q} u_2^i u_3 vw$ and $u_2^{i+1+\lambda K} u_3 vw \approx_1^{R_q} u_2^{i+1} u_3 vw$ for all $\lambda \geq 0$. The authors of [19] seem to assume that $\approx_1$ is a *left-congruence*, i.e. $x \approx_1 y$ implies $ax \approx_1 ay$, which is not true in general.

As a simple fix, one can replace $\approx_1^{R_q}$ by the following left-congruence

$$x \dot{\approx}_1^{R_q} y \iff^{\text{def}} \text{ for all } z \in \Sigma^*: zx \approx_1^{R_q} zy.$$

With this replacement, the verbatim proof in [19] shows that $\approx^{R_q}_1$ has infinite index. It is not difficult to show that this contradicts the assumption made in the lemma that $\approx^R_1$ has finite-index.

B. Proofs for Section VI

Proposition 5. *Given a k-tape automaton $\mathcal{A}$ for a relation R, one can compute in logspace a k-tape automaton $\mathcal{B}$ for a relation S such that R is recognizable if and only if S is synchronous. If $\mathcal{A}$ is deterministic, then so is $\mathcal{B}$.*

Proof. Let $R \subseteq (\Sigma^*)^k$ be a rational relation and $\Sigma' \stackrel{\text{def}}{=} \Sigma \cup \{\#, \vdash\}$ with $\#, \vdash \notin \Sigma$. We show that R is recognizable if and only if

$$R' \stackrel{\text{def}}{=} \{(\#^{n_1} \vdash w_1, \dots, \#^{n_k} \vdash w_k) \mid n_i \geq 0 \text{ and } \boldsymbol{w} \in R\}$$

is synchronous. Since from a k-tape automaton for R we can easily construct a k-tape automaton for R' in logspace while preserving determinism, this shows that recognizability is logspace reducible to synchronicity for both relations in **Rat** and relations in **DRat**.

For the "only if" direction assume that $R = \bigcup_{i=1}^n L_{i,1} \times \cdots \times L_{i,k}$ for regular languages $L_{i,j} \subseteq \Sigma^*$. Then $R' = \bigcup_{i=1}^n \{\#\}^* \{\vdash\} L_{i,1} \times \cdots \times \{\#\}^* \{\vdash\} L_{i,k}$ where the $\{\#\}^* \{\vdash\} L_{i,j} \subseteq (\Sigma')^*$ are clearly regular languages. Thus, R' is recognizable and therefore also synchronous.

For the "if" direction we assume that R is not recognizable which by Proposition 1 means that $\approx_r$ has infinite index for some $r \in [1, k-1]$. Thus, there are words $v_1, v_2, \dots \in \Sigma^*$ and tuples $\boldsymbol{w_{i,j}} = (w_{i,j,1}, \dots, w_{i,j,k-1}) \in (\Sigma^*)^{k-1}$ such that $v_i \odot_r \boldsymbol{w_{i,j}} \in R$ if and only if $v_j \odot_r \boldsymbol{w_{i,j}} \notin R$ for all $i < j$. We show that the Myhill-Nerode equivalence relation $\sim_{\otimes R'}$ of the language of convolutions $\otimes R'$ has infinite index. This implies that $\otimes R'$ cannot be regular and therefore R' is not synchronous. Let

$$x_i \stackrel{\text{def}}{=} \#^{n_i} \vdash \otimes \cdots \otimes \#^{n_i} \vdash \otimes \vdash v_i \otimes \#^{n_i} \vdash \otimes \cdots \otimes \#^{n_i} \vdash$$

with $\vdash v_i$ in the r-th component and $n_i \stackrel{\text{def}}{=} |v_i|$ for all $i \geq 1$ and let

$$y_{i,j} \stackrel{\text{def}}{=} w_{i,j,1} \otimes \cdots \otimes w_{i,j,r-1} \otimes \varepsilon \otimes w_{i,j,r} \otimes \cdots \otimes w_{i,j,k-1}$$

for all $i < j$. Then we have that $x_i y_{i,j} \in \otimes R'$ if and only if $x_j y_{i,j} \notin \otimes R'$ which means that $x_i \not\sim_{\otimes R'} x_j$ for all $i < j$. $\qquad \square$

Lemma 15. *Given $q \in L_t$ and independent multitape automata $\mathcal{I}^{\otimes}_{q'}$ for $R^{\otimes}_{q'}$ for all $q' \in L_{t+1} \cup \cdots \cup L_k$, one can compute $\mathcal{A}^{\otimes}_q$ in time polynomial in the sizes of the $\mathcal{I}^{\otimes}_{q'}$ and exponential in the size of $\mathcal{A}$.*

Proof. Let R be given by a deterministic k-tape automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ with partition of states $Q = Q_1 \uplus \cdots \uplus Q_k$. Let $q \in Q_P$ for a partition $P = \{B_1, \dots, B_t\}$ of $\{1, \dots, k\}$. We show how to construct a deterministic t-tape automaton $\mathcal{A}^{\otimes}_q = (Q', \Sigma', q'_0, \delta', F')$ with $Q' = Q'_1 \uplus \cdots \uplus Q'_t$ and $\Sigma' \stackrel{\text{def}}{=} \Sigma_{\perp}^{|B_1|} \cup \cdots \cup \Sigma_{\perp}^{|B_t|} \cup \{\dashv\}$ where $\Sigma_{\perp} \stackrel{\text{def}}{=} \Sigma \cup \{\perp\}$ and $\perp, \dashv \notin \Sigma$ that

recognizes the relation $\{\boldsymbol{w}(\dashv, \dots, \dashv) \mid \boldsymbol{w} \in R^{\otimes}_q\}$ assuming that we already constructed independent multitape automata $\mathcal{I}^{\otimes}_{q'}$ for $q' \in L_{t+1} \cup \cdots \cup L_k$. The automaton $\mathcal{A}^{\otimes}_q$ consists of two parts. The first part simulates $\mathcal{A}$ over the states in Q_P and the second part simulates the independent multitape automata of states of higher layers than q.

First part: The first part of $\mathcal{A}^{\otimes}_q$ consists of states of the form $(p, \boldsymbol{w})$ with $p \in Q_P$ and $w_i \in \Sigma_{\perp}^{\leq |Q|}$ for $1 \leq i \leq k$. Intuitively, p stores the state of $\mathcal{A}$ the simulation is currently at and $\boldsymbol{w}$ stores a queue for every component of $\mathcal{A}$ of symbols that $\mathcal{A}$ still has to be simulated on. The initial state is $q'_0 \stackrel{\text{def}}{=} (q, \varepsilon, \dots, \varepsilon)$. For input $a' = (a_1, \dots, a_{|B_i|}) \in \Sigma_{\perp}^{|B_i|}$ for $i \in [1, t]$ and $a_j \in \Sigma$ for $j \in [1, |B_i|]$ and state $(p, \boldsymbol{w})$ with $p \in Q_{b_{i,j}}$, $w_{b_{i,j}} = \varepsilon$, and $p' \stackrel{\text{def}}{=} \delta(p, a_j) \in Q_P$ we let $(p, \boldsymbol{w}) \in Q'_i$ and $\delta'((p, \boldsymbol{w}), a') \stackrel{\text{def}}{=} (p', \boldsymbol{w'})$ where

$$w'_{b_{i',j'}} \stackrel{\text{def}}{=} \begin{cases} w_{b_{i,j'}} a_{j'}, & \text{if } i' = i \text{ and } j' \neq j \\ w_{b_{i',j'}}, & \text{otherwise} \end{cases}$$

for all $i' \in [1, t]$ and $j' \in [1, |B_{i'}|]$. For state $(p, \boldsymbol{w})$ with $p \in Q_i$ for some $i \in [1, k]$, $w_i = au$ for some $a \in \Sigma$ and $u \in \Sigma_{\perp}^*$, and $p' \stackrel{\text{def}}{=} \delta(p, a) \in Q_P$ we define $\delta'((p, \boldsymbol{w}), \varepsilon) \stackrel{\text{def}}{=} (p', w_1, \dots, w_{i-1}, u, w_{i+1}, \dots, w_k)$. Note that these ε-transitions can be eliminated without introducing non-determinism since there is no branching possible. On each state of the form $(p, \varepsilon, \dots, \varepsilon)$ with $p \in F$ we append a chain of transitions reading the endmarker $\dashv$ in every component and mark the last state of that chain (which is a sink state) as final.

First to second part: We now define the transitions from states of the first part to states of the second part. Let $p' \in Q_{P'}$ for partition $P' = \{B'_1, \dots, B'_{t'}\}$ that is strictly finer than P and $\mathcal{I}^{\otimes}_{p'} = (\mathcal{A}_1, \dots, \mathcal{A}_{t'}, F_{p'})$ be an independent t'-tape automaton for $R^{\otimes}_{p'}$ with $\mathcal{A}_i = (Q^i, \Sigma_{\perp}^{|B'_i|}, \delta^i, q^i_0)$. The second part consists of states of the form $(q^1, \dots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, m)$ with $q^i \in Q^i$ for $1 \leq i \leq t'$, $\boldsymbol{w}$ is as in the first part, $e_i \in \{0, 1\}$ for $1 \leq i \leq t'$, and $1 \leq m \leq t + 1$. Intuitively, q^i is the state the simulation of $\mathcal{A}_i$ is currently at, e_i stores in a bit whether the simulation of $\mathcal{A}_i$ is finished, and m indicates which component of $\mathcal{A}^{\otimes}_q$ is currently read. For input $a' = (a_1, \dots, a_{|B_i|}) \in \Sigma_{\perp}^{|B_i|}$ for $i \in [1, t]$ and $a_j \in \Sigma$ for $j \in [1, |B_i|]$ and state $(p, \boldsymbol{w})$ with $p \in Q_{b_{i,j}}$, $w_{b_{i,j}} = \varepsilon$, and $\delta(p, a_j) = p'$ we let $(p, \boldsymbol{w}) \in Q'_i$ and $\delta'((p, \boldsymbol{w}), a') \stackrel{\text{def}}{=} (q^1_0, \dots, q^{t'}_0, \boldsymbol{w'}, 0, \dots, 0, 1)$ where

$$w'_{b_{i',j'}} \stackrel{\text{def}}{=} \begin{cases} w_{b_{i,j'}} a_{j'}, & \text{if } i' = i \text{ and } j' \neq j \\ w_{b_{i',j'}}, & \text{otherwise} \end{cases}$$

for all $i' \in [1, t]$ and $j' \in [1, |B_{i'}|]$. For state $(p, \boldsymbol{w})$ with $p \in Q_i$ for some $i \in [1, k]$, $w_i = au$ for some $a \in \Sigma$ and $u \in \Sigma_{\perp}^*$, and $\delta(p, a) = p'$ we define $\delta'((p, \boldsymbol{w}), \varepsilon) \stackrel{\text{def}}{=} (q^1_0, \dots, q^{t'}_0, w_1, \dots, w_{i-1}, u, w_{i+1}, \dots, w_k, 0, \dots, 0, 1)$. Note that these ε-transitions can be eliminated again.

Second part: Let $f \colon \{1, \dots, t'\} \to \{1, \dots, t\}$ with $i' \mapsto i$ such that $B'_{i'} \subseteq B_i$. Note that f is well defined since P' is finer that P. For input $a' = (a_{b_{i,1}}, \dots, a_{b_{i,|B_i|}}) \in \Sigma_{\perp}^{|B_i|} \setminus \{(\perp, \dots, \perp)\}$ for $i \in [1, t]$ and

state $(q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, i)$ we let $(q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, i) \in Q_i'$ and $\delta'((q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, i), a') \stackrel{\text{def}}{=} (p^1, \ldots, p^{t'}, \boldsymbol{w}', \boldsymbol{e}', i)$ where for all $i' \in [1, t']$,

$$p^{i'} \stackrel{\text{def}}{=} \delta^{i'}(q^{i'}, (c_{b'_{i',1}}, \ldots, c_{b'_{i', |B'_{i'}|}}))$$

if $f(i') = i$, $c_{b'_{i',j'}} \neq \bot$ for some j', and $e_{i'} = 0$,

$$p^{i'} \stackrel{\text{def}}{=} q^{i'}$$

if $f(i') \neq i$ or $f(i') = i$ and $c_{b'_{i',j'}} = \bot$ for all j', and otherwise $p^{i'}$ is undefined. Here, we set

$$c_{b'_{i',j'}} \stackrel{\text{def}}{=} \begin{cases} a_{b'_{i',j'}}, & \text{if } w_{b'_{i',j'}} = \varepsilon \\ c, & \text{if } w_{b'_{i',j'}} = cu \text{ for some } c \in \Sigma_\bot, u \in \Sigma_\bot^* \end{cases}$$

for all $i' \in [1, t']$ with $f(i') = i$ and $j' \in [1, |B'_{i'}|]$. Furthermore, we define

$$w'_{b'_{i',j'}} \stackrel{\text{def}}{=} \begin{cases} ua_{b'_{i',j'}}, & \text{if } f(i') = i \text{ and } w_{b'_{i',j'}} = cu \\ w_{b'_{i',j'}}, & \text{otherwise} \end{cases}$$

and

$$e'_{i'} \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } f(i') = i \text{ and } c_{b'_{i',j'}} = \bot \text{ for all } j' \\ e_{i'}, & \text{otherwise} \end{cases}$$

for all $i' \in [1, t']$ and $j' \in [1, |B'_{i'}|]$. For $i \in [1, t]$ and state $(q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, i)$ we let $(q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, i) \in Q_i$ and $\delta'((q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, i), \dashv) \stackrel{\text{def}}{=} (q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, i+1)$. A state of the form $(q^1, \ldots, q^{t'}, \boldsymbol{w}, \boldsymbol{e}, n+1)$ with $w_i = u_i v_i$ for $u_i \in \Sigma^*$ and $v_i \in \{\bot\}^*$ is final if for all $i' \in [1, t']$ we have that $\mathcal{A}_{i'}$ reaches state $f^{i'}$ from $q^{i'}$ on reading $u_{b'_{i',1}} \otimes \cdots \otimes u_{b'_{i', |B'_{i'}|}}$ and $(f^1, \ldots, f^{t'}) \in F_{p'}$. $\qquad \square$