Monadic Decomposability of Regular Relations

Pablo Barceló
Department of Computer Science, University of Chile & IMFD Chile
pbarcelo@dcc.uchile.cl

Chih-Duo Hong
Department of Computer Science, University of Oxford, United Kingdom
chih-duo.hong@st-hughs.ox.ac.uk

Xuan-Bach Le
Department of Computer Science, University of Oxford, United Kingdom
bachdylan@gmail.com

Anthony W. Lin
Technische Universität Kaiserslautern, Germany
anthony.lin@cs.uni-kl.de

Reino Niskanen
Department of Computer Science, University of Oxford, United Kingdom
reino.niskanen@cs.ox.ac.uk

Abstract

Monadic decomposibility — the ability to determine whether a formula in a given logical theory can be decomposed into a boolean combination of monadic formulas — is a powerful tool for devising a decision procedure for a given logical theory. In this paper, we revisit a classical decision problem in automata theory: given a regular (a.k.a. synchronized rational) relation, determine whether it is recognizable, i.e., it has a monadic decomposition (that is, a representation as a boolean combination of cartesian products of regular languages). Regular relations are expressive formalisms which, using an appropriate string encoding, can capture relations definable in Presburger Arithmetic. In fact, their expressive power coincide with relations definable in a universal automatic structure; equivalently, those definable by finite set interpretations in WS1S (Weak Second Order Theory of One Successor). Determining whether a regular relation admits a recognizable relation was known to be decidable (and in exponential time for binary relations), but its precise complexity still hitherto remains open. Our main contribution is to fully settle the complexity of this decision problem by developing new techniques employing infinite Ramsey theory. The complexity for DFA (resp. NFA) representations of regular relations is shown to be NLOGSPACE-complete (resp. PSPACE-complete).

2012 ACM Subject Classification Theory of computation → Regular languages; Theory of computation → Transducers; Theory of computation → Complexity classes; Theory of computation → Logic and verification; Theory of computation → Automated reasoning

Keywords and phrases Transducers, Automata, Synchronized Rational Relations, Ramsey Theory, Variable Independence, Automatic Structures

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.98

Related Version A full version of the paper is available at `https://arxiv.org/abs/1903.00728`.

Funding Barceló is funded by the Millennium Institute for Foundational Research on Data (IMFD) and Fondecyt grant 1170109. Le, Lin, and Niskanen are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no 759969).

Acknowledgements We thank Leonid Libkin for the useful discussion.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 98; pp. 98:1–98:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Monadic decompositions for computable relations have been studied in many different guises, and applied to many different problem domains, e.g., see [17, 25, 38, 12, 27, 28, 37]. The notion of "monadic decomposability" essentially captures the intuitive notion that the components in a given n-ary relation $R \subseteq U^n$ are sufficiently independent from (i.e. not tightly coupled, or interdependent, with) each other. Some examples are in order. Given two subsets $X, Y \subseteq U$, then $X \times Y$ is an instance of relations whose two components are completely independent from each other. On the other hand, the equality relation $\{(x, x) : x \in U\}$ is an example of relations whose two components are tightly coupled. In this paper, we will adopt the commonly studied notion of component-independence[1] (e.g. [25, 38, 7, 37]) in a relation $R \subseteq U^n$ that lies between the extremes as exemplified in the above examples, i.e., that R is expressible as a *finite union* $\bigcup_{i=1}^{r} X_{i,1} \times \cdots \times X_{i,n}$ of products, where each $X_{i,j}$ is expressible in the same language $\mathcal{L}$ (e.g. a logic or a machine model) wherein R is expressed.

Why should one care about monadic decomposable relations? The main reason is that applying appropriate monadic restrictions could make an undecidable problem decidable, and in general turn a difficult problem into one more amenable to analysis. Several examples are in order. Firstly, the well-known cartesian abstractions in abstract interpretation [17] overapproximate the set $R \subseteq U^n$ of reachable states at a certain program point by a relation $R' \subseteq X_1 \times \cdots \times X_m$ such that $R \subseteq R'$. Having R' instead of R sometimes allows a static analysis tool to prove correctness properties about a program that is otherwise difficult to do with only R. Another example includes restrictions to monadic predicates in undecidable logics that result in decidability, e.g., monadic first-order logic and extensions ([9, 10, 4]), as well as monadic second-order theory of successors [10]. Monadic decomposability also found applications in more efficient variable elimination in constraint logic programming (e.g. [23]), as well as constraint processing algorithms for constraint database queries (e.g. [25, 24]). Finally, monadic decompositions in the context of SMT (Satisfiability Modulo Theories), whose study was recently initiated in [38], have numerous applications, including constraint solving over strings [38, 14].

The focus of this paper is to revisit a classical problem of determining monadic decomposability of *regular relations*, which are also known as *synchronized rational relations* [20, 6, 8]. The study of classes of relations over words definable by different classes of multi-tape (finite) automata is by now a well-established subfield of formal language theory. This study was initiated by Elgot, Mezei, and Nivat in the 1960s [18, 30]; also see the surveys [7, 15]. In particular, we have a strict hierarchy of classes of relations as follows: recognizable relations, synchronized rational relations, deterministic rational relations, and rational relations. All these classes over unary relations (i.e. languages) coincide with the class of regular languages. *Rational relations* are relations $R \subseteq (\Sigma^*)^n$ definable by multi-tape automata, where the tape heads move from left to right (in the usual way for finite automata) but possibly at different speeds (e.g. in a transition, the first head could stay at the same position, whereas the second head moves to the right by one position). *Deterministic rational relations* are simply those rational relations that can be described by deterministic multi-tape automata. So far, the heads of the tapes can move at different speeds. *Regular relations* (a.k.a. *synchronized rational relations*) are those relations that are definable by multi-tape automata, all of whose heads move to the right in each transition. Unlike (non)deterministic rational relations, regular relations are extremely well-behaved, e.g., they are closed under first-order operations

[1] Also called variable-independence.

and, therefore, have decidable first-order theories [22]. Regular relations are also known to coincide with those relations that are first-order definable over a universal automatic structure [6, 8]; equivalently, those relations that are definable by finite-set interpretations in the weak-monadic theory of one successor (WS1S) [16]. Finally, the weakest class of relations in the hierarchy are *recognizable relations*: those relations that are definable as a finite union of products of regular languages or, equivalently, relations that can be defined as a boolean combination of regular constraints (i.e. atomic formulas of the form $x \in L$, where L is a regular language, asserting that the word x is in L). Recognizable relations are, therefore, those relations definable by multi-tape automata that exhibit monadic decomposability.

One of the earliest results on deciding whether a relation is monadic decomposable follows from Stearns in 1967 [33] and the characterization of a binary relation $R \subseteq A^* \times B^*$ by $L_R = \{\mathsf{rev}(u)\#v \mid (u, v) \in R\}$, where $\mathsf{rev}(u)$ is the mirror image of u. In [12] it was proven that L_R is a regular language if and only if R has a monadic decomposition and if R is a deterministic rational relation, then L_R is a deterministic context-free language. Due to this characterization, Stearns's result implies that whether a deterministic n-ary rational relation is monadic decomposable (i.e. recognizable) is decidable in the case when $n = 2$. Shortly thereafter, Fischer and Rosenberg [19] showed that the same problem is unfortunately undecidable for the full class of binary rational relations. A few years later Valiant [37] improved the upper bound complexity for the case solved by Stearns to double exponential-time. This is still the best known upper bound for the monadic decomposability problem for deterministic binary rational relations to date and, furthermore, no specific lower bounds are known. More recently Carton *et al.* [12] adapted the techniques from [33, 37] to show that this decidability extends to general n-ary relations, though no complexity analysis was provided. The problem of monadic decomposability for regular relations has also been studied in the literature. Of course decidability with a double exponential-time upper bound for the binary case follows from [37]. In 2000 Libkin [25] gave general conditions for monadic decomposability for first-order theories, which easily implies decidability for monadic decomposability for general k-ary regular relations. This is because regular relations are simply those relations that are definable in a universal automatic structures [6, 8]. The result of Libkin was not widely known in the automata theory community and in fact the problem was posed as an open problem in French version of [31] in 2003 and later on, Carton *et al.* [12] provided a double-exponential-time algorithm for deciding whether an n-ary regular relation is monadic decomposable. More precisely, even though it was claimed in the paper that the algorithm runs in single-exponential time, it was noted in a recent paper by Löding and Spinrath [27, 28] (with which the authors of [12] also agreed, as claimed in [28]) that the algorithm actually runs in double-exponential time. Löding and Spinrath [27, 28] gave a single-exponential-time algorithm (inspired by techniques from [37]) for monadic decomposability of *binary* regular relations.

Contributions

In this paper we provide the precise complexity of monadic decomposability of regular relations, closing the open questions left by Carton *et al.* [12] and Löding and Spinrath [27, 28]. In particular, we show the following.

▶ **Theorem 1.** *Deciding whether a given regular relation R is monadic decomposable is* NLOGSPACE*-complete (resp.* PSPACE*-complete), if R is given by a DFA (resp. an NFA).*

The lower bounds hold already for binary relations (Lemma 5 and Lemma 6 in Section 3). To prove the upper bounds, we first prove the upper bounds for binary relations (Lemma 10 in Section 4) and then extend them to n-ary relations for any given $n > 2$ (Lemma 11 in Section 5).

The existing proof techniques (e.g. in [12, 28, 25]) for deciding monadic decomposability typically aim for finding proofs that the relations are monadic decomposable. In contrast, our proof technique relies on finding a proof that a relation is *not* monadic decomposable. As a brief illustration, suppose we want to show that the regular relation $R = \{(v, v) : v \in \Sigma^*\}$ is not monadic decomposable. We define an equivalence relation $\sim \subseteq \Sigma^* \times \Sigma^*$ as

$$x \sim y := \forall z([R(x, z) \leftrightarrow R(y, z)] \wedge [R(z, x) \leftrightarrow R(z, y)]).$$

This relation is regular since regular relations are closed under first-order operations [31] (a fact that was also used in [12]), but the size of the automaton for this relation is unfortunately quite large; see [27] for detailed discussion. Therefore, we will only use the complement $\not\sim$, which has a substantially smaller representation: polynomial (resp. exponential) size if R is given as a DFA (resp. an NFA). Now, that R is not monadic decomposable amounts to the existence of an ω-sequence $\sigma = \{v_i\}_{i \in \mathbb{N}}$ of words such that $v_i \not\sim v_j$ for each pair $i, j \in \mathbb{N}$. By applying the pigeonhole principle and König's lemma, we will first construct a nicer sequence α (see the top half of Figure 2) and then by exploiting Ramsey Theorem over infinite graphs, we will show that there is an even nicer sequence α' (see the bottom half of Figure 2), where the automaton for $\not\sim$ synchronizes its states in particular points of the computation, no matter which pair of words from the sequence is being read. Moreover, we prove that one of the synchronizing states has a pumping property. This leads to our NLOGSPACE algorithm as we can guess the synchronizing states and verify that there is an accepting run that can be pumped. This technique was inspired by a technique for proving recurrent reachability in regular model checking [34, 35].

The exponential-time upper bound for the binary case from Löding and Spinrath [28] (which is inspired by the techniques used by Stearns [33] and Valiant [37]) relied on characterization of a relation R using the language $L_R = \{\mathsf{rev}(u)\#v \mid (u, v) \in R\}$ and used a suitable machinery that is able to decide whether L_R is regular or not. Their result is not easily extensible to n-ary relations as the encoding of a binary rational relation as a context-free language L_R does not generalize to n-ary relations. In Section 5, we show that proving monadic decomposability for an n-ary regular relation is LOGSPACE-reducible to testing whether linearly many induced binary relations are monadic decomposable.

We conclude in Section 6 with some perspectives from formal verification and a future research direction. The proofs omitted due to length constraints can be found in [5].

2 Preliminaries

A finite alphabet is denoted by Σ and the free monoid it generates by Σ^*. That is, Σ^* consists of all finite words over Σ. The empty word is ε. We denote by $|w|$ the length of word $w \in \Sigma^*$. We have that $|\varepsilon| = 0$. The word $u \in \Sigma^*$ is a *prefix* of $w \in \Sigma^*$ if $w = uv$ for some $v \in \Sigma^*$. We denote this by $u \leq w$. We also write $v = u^{-1}w$, when u is a prefix of w, to state that v is the suffix of w that is obtained after prefix u is removed. Sometimes we want to consider a suffix of w after a prefix of particular length is removed without specifying the actual prefix as defined above. To this end, we define partial function $\sigma : \Sigma^* \times \mathbb{N} \to \Sigma^*$ such that $\sigma(w, i) = v$, where $w = uv$ for some $u \in \Sigma^*$ such that $|u| = i$. In particular, for $u \leq w$, $\sigma(w, |u|) = u^{-1}w$. Similarly, we define partial function $\tau : \Sigma^* \times \mathbb{N} \to \Sigma^*$ such that $\tau(w, i) = u$, where $|u| = i$ and $u \leq w$.

In this paper we study relations $R \subseteq \Sigma^* \times \cdots \times \Sigma^*$ with particular structural properties. Namely, *monadic decomposable* relations that are a finite union of direct products of regular languages, and *regular* relations defined by n-tape finite automata, where the heads move in synchronized manner. See, for example, [31] for more details on such relations.

▶ **Definition 2.** *An n-ary relation $R \subseteq \Sigma^* \times \cdots \times \Sigma^*$ is a* monadic decomposable *relation iff it is of the form $\bigcup_{i=1}^{m}(X_{1,i} \times \cdots \times X_{n,i})$, where m is finite and each $X_{j,i} \subseteq \Sigma^*$ is a regular language.*

As mentioned earlier, this can be intuitively seen as the components of R being independent in some sense. Note that in the literature, monadic decomposable relations are sometimes called *recognizable*. The monadic decomposable relations can be defined using multi-tape automata as is done, e.g., in [12]. The above definition is more suitable for our considerations.

Let $\perp$ be a fresh symbol not found in Σ. We use it to pad words in a relation $R \subseteq \Sigma^* \times \cdots \times \Sigma^*$ in order for each component to be of the same length. Formally, a tuple $(w_1, \ldots, w_n)$ is transformed into $(w_1\perp^{\ell_1}, \ldots, w_n\perp^{\ell_n})$, where $\ell_i = -|w_i| + \max_{1\leq j\leq n}|w_j|$ for each $i = 1, \ldots, n$. We extend this to the relation $R_\perp$ in the expected way. We also denote $\Sigma \cup \{\perp\}$ by $\Sigma_\perp$. An *n-tape automaton over alphabet* $\Sigma_\perp$ is a tuple $(Q, \rightarrow_{\mathcal{A}}, q_0, F)$, where Q is the finite set of states, q_0 is the initial state, F is the set of final states, and $\rightarrow_{\mathcal{A}} \subseteq Q \times (\Sigma_\perp)^n \times \mathcal{P}(Q)$.

▶ **Definition 3.** *An n-ary relation $R \subseteq \Sigma^* \times \cdots \times \Sigma^*$ is* regular *iff $R_\perp$ is recognized by some n-tape automaton $\mathcal{A}_\perp$ over alphabet $\Sigma_\perp$.*

That is, in a regular relation the n heads of the automaton are moving in synchronized manner and the n-tuple of symbols seen determines the state transition. Naturally, the state transition can be deterministic or non-deterministic. We say that a regular relation is defined by an NFA if the underlying n-tape automaton is non-deterministic, otherwise we say that the relation is defined by a DFA. Note that in the literature, regular relations are sometimes called *synchronous rational* or *automatic* relations.

We recall a useful characterization from [12]. Consider an n-ary regular relation $R \subseteq \Sigma^* \times \cdots \times \Sigma^*$. For each $j = 1, \ldots, n-1$, let $\sim_j$ be the following induced equivalence relation:

$$\begin{aligned}(u_1, \ldots, u_j) \sim_j (v_1, \ldots, v_j) \;:=\; &\forall (w_{j+1}, \ldots, w_n) \in \Sigma^* \times \cdots \times \Sigma^* \text{ we have that}\\ &(u_1, \ldots, u_j, w_{j+1}, \ldots, w_n) \in R \iff (v_1, \ldots, v_j, w_{j+1}, \ldots, w_n) \in R \text{ and}\\ &(w_{j+1}, \ldots, w_n, u_1, \ldots, u_j) \in R \iff (w_{j+1}, \ldots, w_n, v_1, \ldots, v_j) \in R.\end{aligned}$$

▶ **Lemma 4** ([12]). *The n-ary regular relation R is monadic decomposable iff $\sim_j$ has finite index for each $j = 1, \ldots, n-1$. That is, there are finitely many equivalence classes over $\sim_j$.*

In other words, R is not monadic decomposable iff for some $j = 1, \ldots, n-1$, there is an infinite sequence $\{u_i\}_{i\geq 0}$, where each u_i is a j-tuple of words, such that for each $0 \leq i < \ell$ it is the case that $u_i \neq u_\ell$ and $u_i \not\sim_j u_\ell$.

In Section 4, we focus on binary relations for which we simplify the notation as there is only one possible value of j. We write $\sim$ instead of $\sim_j$ and $R^{\not\sim}$ for the binary regular relation

$$\begin{aligned}R^{\not\sim}(w, w') \;:=\; \exists u \big(&(R(w,u) \wedge \neg R(w',u)) \,\vee\, (\neg R(w,u) \wedge R(w',u)) \,\vee\\ &(R(u,w) \wedge \neg R(u,w')) \,\vee\, (\neg R(u,w) \wedge R(u,w'))\big).\end{aligned}$$

That is, $R^{\not\sim}$ consists of all words $w, w' \in \Sigma^*$ for which there exists a word $u \in \Sigma^*$ such that one of $R(w,u)$ and $R(w',u)$ is accepted while the other is not, or one of $R(u,w)$ and $R(u,w')$ is accepted while the other is not.

We assume that the reader is familiar with complexity classes and logarithmic space reductions via logarithmic space transducers; see for example [32].

3 Hardness of deciding monadic decomposability of regular relations

In this section, we consider binary regular relations given by NFA and provide a PSPACE lower bound for deciding if such a relation is monadic decomposable. Then, we prove that the same problem for DFA is NLOGSPACE-hard.

▶ **Lemma 5.** *The problem of deciding whether a binary regular relation given by an NFA is monadic decomposable is* PSPACE*-hard.*

Proof. We give a logarithmic space reduction from the universality problem for NFA, which is PSPACE-hard [29]. Recall that in this problem, we are asked to decide whether $L(\mathcal{A}) = \Sigma^*$ given an NFA $\mathcal{A}$ over Σ.

Let $\mathcal{A}$ be an NFA over alphabet Σ, and let $\{\#\}$ be a fresh symbol that we will use as a separator symbol. We assume that $\# \neq \perp$. We construct relation $R = R_1 \cup R_2$ using the language L of $\mathcal{A}$, where

$$R_1 = \{(u, u) \mid u \in (\Sigma \cup \{\#\})^*\} \qquad \text{and} \qquad R_2 = (L \cdot \{\#\})^* \times (\Sigma^* \cdot \{\#\})^*.$$

Intuitively, R_1 contains all pairs (w_1, w_2) such that $w_1 = w_2 = u_0\#u_1\#\cdots\#u_n\#$, where $u_i \in \Sigma^*$, and R_2 contains all pairs (w_1, w_2) such that $w_1 = v_0\#v_1\#\cdots\#v_m\#$, where $v_i \in L$, and $w_2 = u'_0\#u'_1\#\cdots\#u'_n\#$, where $u'_i \in \Sigma^*$. It is easy to construct an NFA that recognizes R in LOGSPACE. Next we show that $L = \Sigma^*$ iff R is monadic decomposable.

Assume first that $L = \Sigma^*$. Then $R_1 \subseteq R_2$, and thus $R = (\Sigma^* \cdot \{\#\})^* \times (\Sigma^* \cdot \{\#\})^*$ which has a trivial monadic decomposition.

For the other direction, assume that R is monadic decomposable, i.e., $R = \bigcup_{i=1}^{n}(A_i \times B_i)$ for some regular languages A_i, B_i. Let $w \in \Sigma^*$. We show that $w \in L$ as well. Consider a set $\{((w\#)^i, (w\#)^i) \mid i = 1, \ldots, n+1\} \subseteq R_1 \subseteq R$. By the pigeonhole principle, there are two elements $((w\#)^j, (w\#)^j)$ and $((w\#)^k, (w\#)^k)$ that belong to the same component of $\bigcup_{i=1}^{n}(A_i \times B_i)$, say to $A_1 \times B_1$. Therefore, $(w\#)^j \in A_1$ and $(w\#)^k \in B_1$, and hence their direct product, $((w\#)^j, (w\#)^k)$, is in $A_1 \times B_1 \subseteq R$. Recall that $R = R_1 \cup R_2$. Clearly, $((w\#)^j, (w\#)^k) \notin R_1$ as the lengths of the two words are different. It follows that $((w\#)^j, (w\#)^k) \in R_2$ and hence $(w\#)^j \in (L \cdot \{\#\})^*$. This implies that $w \in L$. ◀

▶ **Lemma 6.** *The problem of deciding whether a binary regular relation given by a DFA is monadic decomposable is* NLOGSPACE*-hard.*

The proof is straightforward by a reduction from reachability problem for directed acyclic graphs.

4 Deciding monadic decomposability of binary regular relations

In this section we prove our main technical result.

▶ **Lemma 7.** *There is an* NLOGSPACE *algorithm that takes as input an NFA for $R^{\not\perp}$, where R is a binary regular relation, and decides whether R is monadic decomposable.*

We start by defining some notation. We assume any binary regular relation $R^{\not\perp}$ to be given as an NFA with set of states Q. The $R^{\not\perp}$*-type* of a pair (w_1, w_2) of words over Σ is an element of the *transition monoid*. Recall that the transition monoid transforms any given state $q \in Q$ to a set $Q' \subseteq Q$ of states when reading (w_1, w_2). We denote this by $R^{\not\perp}_{w_1,w_2}(q)$ for each $q \in Q$. We write $\mathsf{types}(R^{\not\perp})$ for the set of all $R^{\not\perp}$-types.

Consider an infinite sequence $\{w_i\}_{i\geq 0}$ of words over Σ as defined in Lemma 4. Additionally, we assume that the words in the sequence are of strictly increasing length and that for each $i > 0$ the words w_i and w_{i+1} have a common prefix of length $|w_{i-1}|$. That is, w_i can be written as $\beta_0 \cdots \beta_{i-1}\alpha_i$, where each β_j and α_i is a non-empty word. To simplify notation, we denote $\rho(w_i) = \beta_0 \cdots \beta_i$. That is, $\rho(w_i)$ is of length $|w_i|$ and is a prefix of w_j, for each $0 \leq i < j$. We will show how to construct such sequence in Proposition 8. The words w_i, w_j and w_k are illustrated in the top of Figure 1.

With each pair (i, j), where $i < j$, we associate the following quinary tuple over $\mathsf{types}(R^{\not\sim})$:

$$\mathfrak{C}_{i,j} \;=\; \big(R^{\not\sim}_{w_i,\rho(w_i)},\, R^{\not\sim}_{\rho(w_i),\rho(w_i)},\, R^{\not\sim}_{\sigma(w_j,|w_i|),\sigma(\rho(w_j),|w_i|)},\, R^{\not\sim}_{\varepsilon,\sigma(w_j,|w_i|)},\, R^{\not\sim}_{\varepsilon,\sigma(\rho(w_j),|w_i|)} \big).$$

Intuitively, the first component corresponds to the computation of $(\beta_0 \cdots \beta_{i-1}\alpha_i, \beta_0 \cdots \beta_{i-1}\beta_i)$, the second to $(\beta_0 \cdots \beta_{i-1}\beta_i, \beta_0 \cdots \beta_{i-1}\beta_i)$ needed in order to compute the third component, $(\beta_{i+1} \cdots \beta_{j-1}\alpha_j, \beta_{i+1} \cdots \beta_{j-1}\beta_j)$. The final two components are used to compute the set of states reachable after the whole word in the first component is read. That is $(\perp^{|\beta_{i+1}\cdots\beta_{j-1}\alpha_j|}, \beta_{i+1} \cdots \beta_{j-1}\alpha_j)$ and $(\perp^{|\beta_{i+1}\cdots\beta_{j-1}\beta_j|}, \beta_{i+1} \cdots \beta_{j-1}\beta_j)$. See Figure 1 for a pictorial depiction.

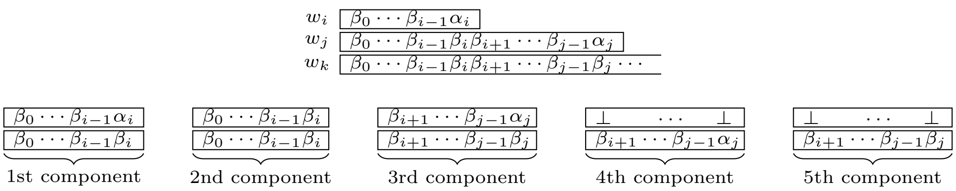

Figure 1 Correspondence between components of $\mathfrak{C}_{i,j}$ and parts of computation on w_i, w_j and w_k, where $i < j < k$.

We can then establish the following important proposition. Consider an infinite sequence of words that are pairwise from different equivalence classes as in Lemma 4. We show next that we can extract an infinite subsequence with additional structural properties. Perhaps the most important property is that $\mathfrak{C}_{i,j}$ is the same for all i, j. This subsequence will allow us to prove the main lemma.

▶ **Proposition 8.** *A binary regular relation R over $\Sigma^* \times \Sigma^*$ is not monadic decomposable iff there are infinite sequences $\{u_i\}_{i\geq 0}$, $\{\gamma_i\}_{i\geq 0}$, and $\{\delta_i\}_{i\geq 0}$ of words over Σ and a quinary tuple $\mathfrak{C}$ over* $\mathsf{types}(R^{\not\sim})$ *such that for each $i \geq 0$ it is the case that*

1. $|\gamma_i| = |\delta_i| > 0$,
2. $u_i = \delta_0 \cdots \delta_{i-1}\gamma_i$,
3. $(u_i, u_j) \in R^{\not\sim}$, *for each $j > i$, and*
4. $\mathfrak{C}_{i,j} = \mathfrak{C}$, *for each $j > i$.*

Proof. By Lemma 4, the existence of such sequences directly implies that the relation is not monadic decomposable. Assume then that R is not monadic decomposable. By Lemma 4, there exists a sequence $\{v_i\}_{i\geq 0}$ such that $R^{\not\sim}(v_j, v_\ell)$ for all $j \neq \ell$. It remains to show how to construct the three sequences satisfying the additional properties from $\{v_i\}_{i\geq 0}$. First, we construct an auxiliary sequence $\{w_i\}_{i\geq 0}$ in the following way. Let v_j be the first non-empty word of $\{v_i\}_{i\geq 0}$. Denote $v_j = w'_0 = \alpha_0$. Consider prefixes of v_i of length $|\alpha_0|$. Since $|\alpha_0|$ is finite and the sequence is infinite, there exists a prefix that appears infinitely often by the pigeonhole principle. Denote this prefix by β_0. Now we consider an infinite subsequence $\{w'_i\}_{i\geq 0}$ of $\{v_i\}_{i\geq 0}$ where $w'_0 = v_j$ and w'_i, where $i > 0$, has β_0 as the proper prefix. We can write $w'_1 = \beta_0\alpha_1$ and repeat the procedure. By König's Lemma, we can always repeat the procedure and obtain the desired auxiliary sequence $\{w_i\}_{i\geq 0}$ in the limit.

From Infinite Ramsey's Theorem, there is an infinite sequence $0 \leq \ell_0 < \ell_1 < \cdots$ and a tuple $\mathfrak{C} \in \mathsf{types}(R^{\not\sim})^5$ such that for each $0 \leq i < j$ we have $\mathfrak{C}_{\ell_i,\ell_j} = \mathfrak{C}$. Namely, we consider a complete infinite graph with natural numbers as vertices. An edge between vertices i and j is coloured with $\mathfrak{C}_{i,j} \in \mathsf{types}(R^{\not\sim})^5$. Now there is an infinite clique coloured with $\mathfrak{C}$ which gives us our infinite sequence $0 \leq \ell_0 < \ell_1 < \cdots$.

We then define the u_is, γ_is, and δ_is, for $i \geq 0$, as follows.

- $\gamma_0 = w_{\ell_0}$ and γ_{i+1}, for $i > 0$, is the word $\sigma(w_{\ell_{i+1}}, |w_{\ell_i}|)$.
- δ_i is defined as $\rho(\gamma_i)$.
- $u_i = \delta_0 \cdots \delta_{i-1}\gamma_i$, for each $i \geq 0$.

It is easy to see then that $u_i = w_{\ell_i}$ and $\rho(u_i) = \delta_0 \cdots \delta_{i-1}\delta_i = \rho(w_{\ell_i})$, for each $i \geq 0$. Therefore, $\{u_i\}_{i\geq 0}$, $\{\gamma_i\}_{i\geq 0}$, $\{\delta_i\}_{i\geq 0}$, and $\mathfrak{C}$ satisfy the conditions in the statement of the proposition. See Figure 2 for a pictorial depiction of the construction. ◀

In other words, by Proposition 8, there is a sequence $\{u_i\}_{i\geq 0}$ and a $\mathfrak{C}$ such that for each i, j, the runs on $R^{\not\sim}$ are synchronized after (γ_i, δ_i), (δ_i, δ_i), $(\delta_i^{-1}\gamma_j, \delta_i^{-1}\delta_j)$, $(\varepsilon, \delta_i^{-1}\gamma_j)$ and $(\varepsilon, \delta_i^{-1}\delta_j)$ have been read. In particular, the runs are synchronized in states of $R^{\not\sim}_{\gamma_i,\delta_i}$, $R^{\not\sim}_{\delta_i,\delta_i}$, $R^{\not\sim}_{\delta_i^{-1}\gamma_j,\delta_i^{-1}\delta_j}$, $R^{\not\sim}_{\varepsilon,\delta_i^{-1}\gamma_j}$ and $R^{\not\sim}_{\varepsilon,\delta_i^{-1}\delta_j}$, respectively.

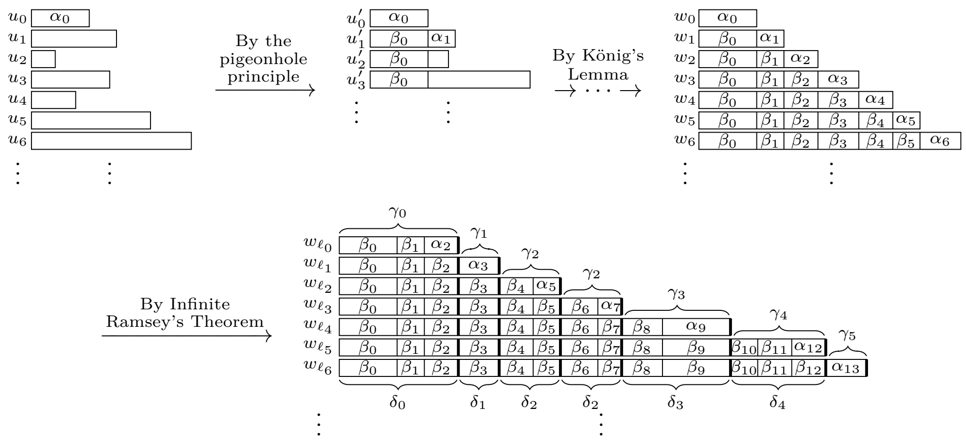

Figure 2 An illustration of construction of sequence $\{u_i\}_{i\geq 0}$ of Proposition 8 in two steps. Here $R^{\not\sim}(u_i, u_j)$, $R^{\not\sim}(u'_i, u'_j)$ and $R^{\not\sim}(w_i, w_j)$ for every $i \neq j$. Moreover as $\mathfrak{C} = \mathfrak{C}_{i,j}$, the sets of states reachable after each δ_i and γ_i are the same (indicated by thick lines).

We can then prove the following crucial result. We assume here that R is a binary regular relation over $\Sigma \times \Sigma$ such that $R^{\not\sim}$ is given as an NFA over $\Sigma \times \Sigma$ whose set of states is Q. We further assume that q_0 is the initial state of $R^{\not\sim}$ and F its set of final states.

▶ **Lemma 9.** *Relation R is not monadic decomposable iff there are an infinite sequence $\{(x_i, y_i)\}_{i\geq 0}$ of pairs of words over Σ and states $q, q', p, r \in Q$, such that $p \in F$, it is the case that $q \in R^{\not\sim}_{x_0,y_0}(q_0)$, and the following statements hold for each $i \geq 0$.*

1. *$|x_i| = |y_i|$ and y_i is a prefix of both x_{i+1} and y_{i+1}.*
2. *$q' \in R^{\not\sim}_{y_i,y_i}(q_0)$; $\quad q \in R^{\not\sim}_{y_i^{-1}x_{i+1},y_i^{-1}y_{i+1}}(q')$; $\quad p \in R^{\not\sim}_{\varepsilon,y_i^{-1}x_{i+1}}(q)$; $\quad r \in R^{\not\sim}_{\varepsilon,y_i^{-1}y_{i+1}}(q)$.*
3. *If $i > 0$, we also have that $p \in R^{\not\sim}_{\varepsilon,y_i^{-1}x_{i+1}}(r)$ and $r \in R^{\not\sim}_{\varepsilon,y_i^{-1}y_{i+1}}(r)$.*

Proof. Assume first that R is not monadic decomposable. By Proposition 8, there are infinite sequences $\{u_i\}_{i\geq 0}$, $\{\gamma_i\}_{i\geq 0}$, and $\{\delta_i\}_{i\geq 0}$ of words over Σ and a quinary tuple $\mathfrak{C}$ over $\mathsf{types}(R^{\not\sim})$ such that for each $i \geq 0$ it is the case that

1. $|\gamma_i| = |\delta_i| > 0$,
2. $u_i = \delta_0 \cdots \delta_{i-1}\gamma_i$,
3. $(u_i, u_j) \in R^{\not\sim}$, for each $j > i$, and
4. $\mathfrak{C}_{i,j} = \mathfrak{C}$, for each $j > i$.

We then define a sequence $\{(x_i, y_i)\}_{i\geq 0}$ such that $x_i := u_i$, for each $i \geq 0$, and y_i is the prefix of $x_{i+1} = u_{i+1}$ that has the same length as $x_i = u_i$, i.e., $y_i = \tau(x_{i+1}, |x_i|)$. Hence, $y_i = \rho(u_i) = \delta_0 \cdots \delta_i$. Clearly, $|x_i| = |y_i| \geq 0$ and y_i is a prefix of both x_{i+1} and y_{i+1}, for each $i \geq 0$. We prove next that the sequence $\{(x_i, y_i)\}_{i\geq 0}$ also satisfies the remaining conditions.

Before defining $q, q', p, r \in Q$, let us highlight the intuition why such states exist for every i. We can find such states because by our assumption $\mathfrak{C}_{i,j} = \mathfrak{C}$ for each $i < j$. Further, whether q is reachable from q_0 is stored in the first component of $\mathfrak{C}$. Similarly, the second and third components of $\mathfrak{C}$ allow us to find q' that is reachable from q_0 and such that q is reachable from q'. Finally, the fourth component is for checking whether p is reachable from q and r, while the fifth component for checking that r is reachable from both q and r.

Let us define $q, q', p, r \in Q$ as follows.

- q and p are states such that $p \in F$ and it is the case that $q \in R^{\not\sim}_{x_0,y_0}(q_0)$ and $p \in R^{\not\sim}_{\varepsilon, y_0^{-1}x_1}(q)$. Notice that such q and p must exist as $(x_0, x_1) \in R^{\not\sim}$, i.e., it holds that $R^{\not\sim}_{x_0,x_1}(q_0) \cap F \neq \emptyset$, and $R^{\not\sim}_{x_0,x_1}(q_0) = R^{\not\sim}_{x_0,y_0}(q_0) \circ R^{\not\sim}_{\varepsilon,y_0^{-1}x_1}$.
- q' is a state such that $q' \in R^{\not\sim}_{y_0,y_0}(q_0)$ and $q \in R^{\not\sim}_{y_0^{-1}x_1, y_0^{-1}y_1}(q')$. Notice that such a q' must exist. Indeed, since $\mathfrak{C}_{0,1} = \mathfrak{C}_{1,2} = \mathfrak{C}$, we have $R^{\not\sim}_{u_0,\rho(u_0)} = R^{\not\sim}_{x_0,y_0} = R^{\not\sim}_{u_1,\rho(u_1)} = R^{\not\sim}_{x_1,y_1}$. This implies that $q \in R^{\not\sim}_{x_1,y_1}(q_0) = R^{\not\sim}_{y_0,y_0}(q_0) \circ R^{\not\sim}_{y_0^{-1}x_1, y_0^{-1}y_1}$, as we know that $q \in R^{\not\sim}_{x_0,y_0}(q_0)$ and there must be an intermediate state q' that is reached after reading (y_0, y_0).
- We have that r is a state such that

$$r \in R^{\not\sim}_{\varepsilon, y_0^{-1}y_1}(q); \qquad p \in R^{\not\sim}_{\varepsilon, y_1^{-1}x_2}(r); \qquad \text{and} \qquad r \in R^{\not\sim}_{\varepsilon, y_1^{-1}y_2}(r).$$

  The existence of such state r is not obvious but straightforward; see [5].

We now prove that q, q', p, r satisfy all the requirements in the statement of the Lemma. By definition, $q \in R^{\not\sim}_{x_0,y_0}(q_0)$ and $p \in F$. We can then prove by induction that for each $i \geq 0$ it is the case that

$$q' \in R^{\not\sim}_{y_i,y_i}(q_0); \quad q \in R^{\not\sim}_{y_i^{-1}x_{i+1}, y_i^{-1}y_{i+1}}(q'); \quad p \in R^{\not\sim}_{\varepsilon, y_i^{-1}x_{i+1}}(q); \quad r \in R^{\not\sim}_{\varepsilon, y_i^{-1}y_{i+1}}(q);$$

and, in addition, that for each $i > 0$ it is the case that $p \in R^{\not\sim}_{\varepsilon, y_i^{-1}x_{i+1}}(r)$ and $r \in R^{\not\sim}_{\varepsilon, y_i^{-1}y_{i+1}}(r)$. The base case $i = 0$ holds by definition. The inductive case is straightforward.

Let us assume now that there are an infinite sequence $\{(x_i, y_i)\}_{i\geq 0}$ of pairs of words over Σ and states $q, q', p, r \in Q$ that satisfy the conditions stated in the statement of the lemma. We prove next that R is not monadic decomposable by showing that there are infinite sequences $\{w_i\}_{i\geq 0}$, $\{\alpha_i\}_{i\geq 0}$ and $\{\beta_i\}_{i\geq 0}$ of words over Σ such that $\{w_i\}_{i\geq 0}$, $\{\alpha_i\}_{i\geq 0}$, and $\{\beta_i\}_{i\geq 0}$ satisfy the conditions stated in Lemma 4.

We define $w_i := x_i$ for each $i \geq 0$. Furthermore, $\alpha_0 := x_0$, $\beta_0 := y_0$, and for each $i > 0$ we set $\alpha_i := y_{i-1}^{-1}x_i$ and $\beta_i := y_{i-1}^{-1}y_i$. Clearly $|\alpha_i| = |\beta_i| > 0$ and $w_i = x_i = \beta_0 \cdots \beta_{i-1}\alpha_i$, for each $i \geq 0$. We prove next that $(w_i, w_j) \in R^{\not\sim}$ for each $0 \leq i < j$. Actually, we prove a stronger claim: $p \in R^{\not\sim}_{w_i,w_j}(q_0)$ and $r \in R^{\not\sim}_{w_i,\rho(w_j)}(q_0)$, for each $0 \leq i < j$, where as before $\rho(w_j) = \tau(w_{j+1}, |w_j|) = \beta_0\beta_1 \cdots \beta_j$. The claim can be proved by induction. ◀

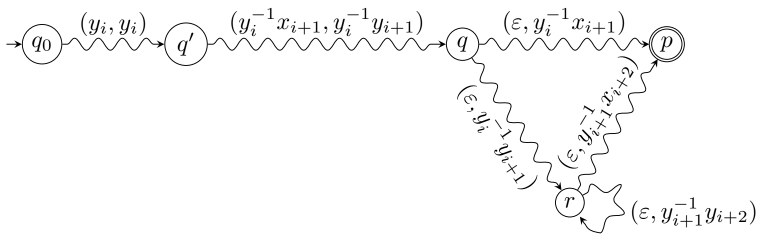

Figure 3 Runs in $R^{\not\sim}$ on states q, q', p and r as defined in Lemma 9. The runs exist for every $i \geq 0$.

The runs as extracted from the sequence $\{(x_i, y_i\})_{i\geq 0}$ satisfying the conditions of Lemma 9 are depicted in Figure 3.

Lemma 9 allows us to reduce the monadic decomposability problem to a set of reachability checks on types. With the help of this property, we can then prove Lemma 7.

Proof of Lemma 7. For each $(q, q', p, r) \in Q \times Q \times Q \times Q$ with $p \in F$ do the following.

- Check if there are words w_0, v_0, w_1, v_1 such that $|w_0| = |v_0| > 0$, $|w_1| = |v_1| > 0$, and it holds that (i) $q \in R^{\not\sim}_{w_0,v_0}(q_0)$, (ii) $q' \in R^{\not\sim}_{v_0,v_0}(q_0)$, (iii) $q \in R^{\not\sim}_{w_1,v_1}(q')$, (iv) $q' \in R^{\not\sim}_{v_1,v_1}(q')$, (v) $p \in R^{\not\sim}_{\varepsilon,w_1}(q)$, and (vi) $r \in R^{\not\sim}_{\varepsilon,v_1}(q)$.
- Check if there are words w, v such that $|w| = |v| > 0$, and it holds that (i) $q \in R^{\not\sim}_{w,v}(q')$, (ii) $q' \in R^{\not\sim}_{v,v}(q')$, (iii) $p \in R^{\not\sim}_{\varepsilon,w}(q)$, (vi) $r \in R^{\not\sim}_{\varepsilon,v}(q)$, (v) $p \in R^{\not\sim}_{\varepsilon,w}(r)$, and (vi) $r \in R^{\not\sim}_{\varepsilon,v}(r)$.

If this holds for any such a tuple, then R is not monadic decomposable. Else, R is monadic decomposable. It is easy to see that this algorithm can be implemented in NLOGSPACE. ◀

We have the necessary ingredients to prove a part of Theorem 1.

▶ **Lemma 10.** *Deciding whether a given binary regular relation R is monadic decomposable is in* NLOGSPACE *(resp. in* PSPACE*), if R is given by a DFA (resp. an NFA).*

Proof. The claim follows from Lemma 7. Namely, from the definition of $R^{\not\sim}$, it follows that, if R is given by a DFA, then $R^{\not\sim}$ can be constructed in LOGSPACE. Indeed, this can be done as disjunctions, conjunctions and projections can all be done in LOGSPACE and then via composability of LOGSPACE transducers we can construct $R^{\not\sim}$ of logarithmic size. (Note that the output of a LOGSPACE transducer is of at most polynomial size.) Then by Lemma 7, we obtain the decidability of monadic decomposability in NLOGSPACE for R given by a DFA.

Similarly, if R is given by an NFA, we construct $R^{\not\sim}$ of polynomial size since an NFA can be transformed into a DFA using a PSPACE transducer. (Again, the output of a PSPACE transducer is of at most exponential size.) Thus monadic decomposability is in PSPACE. ◀

5 Deciding monadic decomposability of regular relations

In this section, we finish the proof of Theorem 1. The remaining component is showing that monadic decomposability of n-ary regular relations is decidable in NLOGSPACE for DFA and PSPACE for NFA.

▶ **Lemma 11.** *Deciding whether a given n-ary regular relation R is monadic decomposable is in* NLOGSPACE *(resp. in* PSPACE*), if R is given by a DFA (resp. an NFA).*

Proof of Theorem 1. The upper bounds follow from Lemma 11 and the lower bound follows from Lemma 5 for NFA and from Lemma 6 for DFA. ◀

In order to prove Lemma 11, we extend Lemma 10 to n-ary relations. Let us first define some helpful notation used throughout the section.

Recall that words of regular relations are padded to be of the same length using $\perp$. We denote this function by $\mathrm{PAD}_\perp$. For example, $\mathrm{PAD}_\perp((a, \varepsilon, ab)) = (a\perp, \perp\perp, ab)$. Let us now define a padding function δ_n that acts slightly differently. Instead of padding the words in a tuple to make them of the same length, the new function pads a sequence of tuples with tuples where some elements are $\perp$. Let us describe δ_n in more details. Define $\Sigma_n = (\Sigma_\perp)^n \setminus \{\perp^n\}$, i.e., an alphabet consisting of n-tuples of letters from $\Sigma_\perp$, excluding $(\perp, \dots, \perp)$. Now $\delta_n : (\Sigma^*)^n \to \Sigma_n^*$ is an injective mapping that uses $\perp$ to extend the shorter words to the same length as the longest word. For example, δ_3 maps $(a, \varepsilon, ab) \in (\Sigma^*)^3$ to $(a, \perp, a)(\perp, \perp, b) \in \Sigma_3^*$ as follows:

$$(a, \varepsilon, ab) \to \begin{pmatrix} a \\ \varepsilon \\ ab \end{pmatrix} \to \begin{pmatrix} a\perp \\ \perp\perp \\ ab \end{pmatrix} \to \begin{pmatrix} a \\ \perp \\ a \end{pmatrix} \begin{pmatrix} \perp \\ \perp \\ b \end{pmatrix} \to (a, \perp, a)(\perp, \perp, b).$$

▶ **Lemma 12.** *For $n \geq 1$, $\{(x_1, \dots, x_n, y) \mid \delta_n(x_1, \dots, x_n) = y\} \subseteq (\Sigma^*)^n \times \Sigma_n^*$ is regular.*

Given an n-ary relation $R \subseteq (\Sigma^*)^n$ and positive integers $x_1, \dots, x_m$ such that $\sum_{i=1}^m x_i = n$, an m-ary relation $R_{x_1,\dots,x_m} \subseteq \Sigma_{x_1}^* \times \cdots \times \Sigma_{x_m}^*$ can be uniquely determined via the mappings $\delta_{x_1}, \dots, \delta_{x_m}$. More precisely, there exists a one-to-one correspondence $\Delta_{x_1,\dots,x_m}$ between relations R and $R_{x_1,\dots,x_m}$ that maps each $(w_1, \dots, w_n) \in R$ to

$$(\delta_{x_1}(w_1, \dots, w_{x_1}), \delta_{x_2}(w_{x_1+1}, \dots, w_{x_1+x_2}), \dots, \delta_{x_m}(w_{x_1+\cdots+x_{m-1}+1}, \dots, w_n)) \in R_{x_1,\dots,x_m}.$$

For example, a ternary relation $R = \{(a, \varepsilon, ab)\}$ over $(\Sigma^*)^3$ uniquely determines a binary relation $R_{1,2} = \{(a, (\perp, a)(\perp, b))\}$ over $\Sigma_1^* \times \Sigma_2^*$ through the correspondence $\Delta_{1,2}$. For the sake of readability, if the integers $x_1, \dots, x_m$ have a constant subsequence of length k, i.e., $x_i = x_{i+1} = \cdots = x_{i+k-1}$ for some i, we write the relation as $R_{x_1,\dots,x_{i-1},x_i^k,x_{i+k},\dots,x_m}$.

In the following, we shall use R_k to denote the binary relation $R_{k,n-k}$ induced by R. It turns out that being able to check monadic decomposability for binary relations is sufficient to check monadic decomposability for general n-ary relations.

▶ **Lemma 13.** *Let R be an n-ary regular relation and let $R_1, \dots, R_{n-1}$ be the induced binary relations. Then R is monadic decomposable iff $R_1, \dots, R_{n-1}$ are monadic decomposable.*

Proof. Define $\delta_i(S) = \{\delta_i(s_1, \dots, s_i) \mid (s_1, \dots, s_i) \in S\}$. The only-if part of the lemma is immediate, since $R = \bigcup_i X_{i,1} \times \cdots \times X_{i,n}$ implies that $R_k = \bigcup_i \delta_k(X_{i,1} \times \cdots \times X_{i,k}) \times \delta_{n-k}(X_{i,k+1} \times \cdots \times X_{i,n})$ for $1 \leq k \leq n-1$, namely, $R_1, \dots, R_{n-1}$ are monadic decomposable.

To see the other direction, we say that an n-ary relation R is *k-decomposable* if the induced k-ary relation $R_{1^{k-1},n-k+1}$ of R is monadic decomposable. Now it suffices to show that R is n-decomposable since $R = R_{1^n}$. We shall prove this by induction on $k \in \{2, \dots, n\}$. Note that R is 2-decomposable by the assumption that R_1 is monadic decomposable. For $2 \leq k \leq n-1$, suppose that $R_k = \bigcup_j A_j \times B_j$ and R is k-decomposable, say $R_{1^{k-1},n-k+1} = \bigcup_i X_{i,1} \times \cdots \times X_{i,k-1} \times Y_i$. Then R is $(k+1)$-decomposable as we have

$$R_{1^k,n-k} = \bigcup_i \bigcup_j X_{i,1} \times \cdots \times X_{i,k-1} \times A_{i,j} \times B_j,$$

where $A_{i,j} = \{x \in \Sigma^* \mid \exists x_1 \in X_{i,1} \cdots \exists x_{k-1} \in X_{i,k-1}.\ \delta_k(x_1, \dots, x_{k-1}, x) \in A_j\}$, i.e., $A_{i,j}$ is the projection of $\delta_k^{-1}(A_j) \cap (X_{i,1} \times \cdots \times X_{i,k-1} \times \Sigma^*)$ on the k-th component. Note that $\delta_k^{-1}(A_j)$ is regular since A_j and $\{(x_1, \dots, x_k, y) \mid \delta_k(x_1, \dots, x_k) = y\}$ are regular (cf. [8]). Hence $A_{i,j}$ is also regular. The claim that R is n-decomposable then follows by induction. ◀

Proof (sketch) of Lemma 11. To prove the lemma, we show that if R is regular, then so are the induced relations $R_1, \ldots, R_{n-1}$. Moreover, given the automaton of R, one can construct the automaton for each R_i in logarithmic space from R. We then check if each R_i is monadic decomposable for $i = 1, \ldots, n-1$. From Lemma 10 the latter is in NLOGSPACE (resp. PSPACE), and thus the whole procedure is in NLOGSPACE (resp. PSPACE) if R is given by a DFA (resp. an NFA). ◀

6 Concluding Remarks

Monadic decomposability for rational relations (and subclasses thereof) is a classical problem in automata theory that dates back to the late 1960s (the work of Stearns [33] and Fischer and Rosenberg [19]). While the general problem is undecidable, the subcase of regular relations (i.e. those recognized by synchronized multi-tape automata) provides a good balance between decidability [25, 12] and expressiveness. The complexity of this subcase remained open for over a decade (exponential-time upper bound for the binary case [27, 28], double exponential-time upper bound in the general case [12], and no specific lower bounds). This paper closes this question by providing the precise complexity for the problem: NLOGSPACE (resp. PSPACE) for DFA (resp. NFA) representations.

Some perspectives from formal verification and future work: Researchers from the area of formal verification have increasingly understood the importance of the monadic decompositions techniques, e.g., see [38]. Directly pertinent to monadic decomposability of regular relations is the line of work of constraint solving over strings, wherein increasingly more complex string operations are needed and thus added to solvers [36, 3, 26, 1, 13, 2, 14]. As an example, let us take a look at the recent work of Chen *et al.* [14], which spells out a string constraint language with semantic conditions for decidability that directly use the notion of monadic decomposability of relations over strings. Loosely speaking, a constraint is simply a sequence of program statements, each being either an assignment or a conditional:

$$S ::= \qquad y := f(x_1, \ldots, x_r) \mid \mathbf{assert}(g(x_1, \ldots, x_r)) \mid S; S$$

where $f : (\Sigma^*)^r \to \Sigma^*$ is a partial string function and $g \subseteq (\Sigma^*)^r$ is a string relation. The meaning of a constraint is what one would expect in a program written in a standard imperative programming language, which should support assignments and assertions. Note that loops are not allowed in the language since their target application is symbolic executions (e.g. see [11]). They provided two semantic conditions for ensuring decidability, one of which requires that each conditional g is effectively monadic decomposable. There is evidence (e.g. [21, 14]) that some form of length reasoning in g is indeed required for many applications of symbolic executions of string-manipulating programs, but much of the length constraints could be (not yet fully automatically) translated to regular constraints. A potential application for our results is therefore to provide support for complex string relations for g in the form of regular relations, which permit a rather expressive class of conditionals (e.g. some form of length reasoning, etc.). Despite this, this application also highlights what is currently missing in the entire literature of monadic decomposability of rational relations: a study of the problem of *outputting* the monadic decompositions of the relations, if monadic decomposable. (In fact, this is also true of other logical theories before the recent work of Veanes *et al.* [38].) What is the complexity of this problem with various representations of recognizable relations (e.g. finite unions of products, boolean combinations of regular constraints, etc.)? Although our results provide *a first step* towards solving this function problem, we strongly believe this to be a highly challenging open problem in its own right that deserves more attention.

References

1. Parosh Aziz Abdulla, Mohamed Faouzi Atig, Yu-Fang Chen, Bui Phi Diep, Lukás Holík, Ahmed Rezine, and Philipp Rümmer. Flatten and conquer: a framework for efficient analysis of string constraints. In *Proceedings of PLDI 2017*, pages 602–617. ACM, 2017. doi:10.1145/3062341.3062384.
2. Parosh Aziz Abdulla, Mohamed Faouzi Atig, Yu-Fang Chen, Bui Phi Diep, Lukás Holík, Ahmed Rezine, and Philipp Rümmer. TRAU: SMT solver for string constraints. In *Formal Methods in Computer Aided Design, FMCAD 2018*, 2018.
3. Parosh Aziz Abdulla, Mohamed Faouzi Atig, Yu-Fang Chen, Lukás Holík, Ahmed Rezine, Philipp Rümmer, and Jari Stenman. String constraints for verification. In *Proceedings of CAV 2014*, volume 8559 of *LNCS*, pages 150–166. Springer, 2014. doi:10.1007/978-3-319-08867-9_10.
4. James Bailey, Guozhu Dong, and Anthony Widjaja To. Logical queries over views: Decidability and expressiveness. *ACM Trans. Comput. Log.*, 11(2):8:1–8:35, 2010. doi:10.1145/1656242.1656243.
5. Pablo Barceló, Chih-Duo Hong, Xuan-Bach Le, Anthony W. Lin, and Reino Niskanen. Monadic decomposability of regular relations. *CoRR*, abs/1903.00728, 2019. URL: https://arxiv.org/abs/1903.00728.
6. Michael Benedikt, Leonid Libkin, Thomas Schwentick, and Luc Segoufin. Definable relations and first-order query languages over strings. *J. ACM*, 50(5):694–751, 2003. doi:10.1145/876638.876642.
7. Jean Berstel. *Transductions and Context-Free Languages*. Teubner-Verlag, 1979.
8. Achim Blumensath. *Automatic Structures*. PhD thesis, RWTH Aachen, 1999.
9. George S. Boolos, John P. Burgess, and Richard C. Jeffrey. *Computability and Logic*. Cambridge University Press, fifth edition, 2007.
10. Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Springer, 1997.
11. Cristian Cadar and Koushik Sen. Symbolic execution for software testing: Three decades later. *Commun. ACM*, 56(2):82–90, 2013. doi:10.1145/2408776.2408795.
12. Olivier Carton, Christian Choffrut, and Serge Grigorieff. Decision problems among the main subfamilies of rational relations. *RAIRO – Theoretical Informatics and Applications*, 40(2):255–275, 2006. doi:10.1051/ita:2006005.
13. Taolue Chen, Yan Chen, Matthew Hague, Anthony W. Lin, and Zhilin Wu. What is decidable about string constraints with the ReplaceAll function. *PACMPL*, 2(POPL):3:1–3:29, 2018. doi:10.1145/3158091.
14. Taolue Chen, Matthew Hague, Anthony W. Lin, Philipp Rümmer, and Zhilin Wu. Decision procedures for path feasibility of string-manipulating programs with complex operations. *PACMPL*, 3(POPL):49:1–49:30, 2019. doi:10.1145/3290362.
15. Christian Choffrut. Relations over words and logic: A chronology. *Bull. of the EATCS*, 89:159–163, 2006.
16. Thomas Colcombet and Christof Löding. Transforming structures by set interpretations. *Logical Methods in Computer Science*, 3(2), 2007. doi:10.2168/LMCS-3(2:4)2007.
17. Patrick Cousot and Radhia Cousot. Systematic design of program analysis frameworks. In *Proceedings of POPL 1979*, pages 269–282, 1979. doi:10.1145/567752.567778.
18. Calvin C. Elgot and Jorge E. Mezei. On relations defined by generalized finite automata. *IBM J. Res. Dev.*, 9(1):47–68, 1965. doi:10.1147/rd.91.0047.
19. Patrick C. Fischer and Arnold L. Rosenberg. Multitape one-way nonwriting automata. *J. Comput. Syst. Sci.*, 2(1):88–101, 1968. doi:10.1016/S0022-0000(68)80006-6.
20. Christiane Frougny and Jacques Sakarovitch. Synchronized rational relations of finite and infinite words. *Theor. Comput. Sci.*, 108(1):45–82, 1993. doi:10.1016/0304-3975(93)90230-Q.
21. Vijay Ganesh, Mia Minnes, Armando Solar-Lezama, and Martin C. Rinard. Word equations with length constraints: What's decidable? In *Proceedings of HVC 2012*, pages 209–226. Springer, 2012. doi:10.1007/978-3-642-39611-3_21.

22 Bernard R. Hodgson. Decidabilite par automate fini. *Ann. Sc. Math. Quebec*, 7:39–57, 1983.

23 Jean-Louis Imbert. Redundancy, variable elimination and linear disequations. In *Proceedings of ILPS 1994*, pages 139–153, 1994.

24 Gabriel Kuper, Leonid Libkin, and Jan Paredaens. *Constraint Databases*. Springer Publishing Company, Incorporated, first edition, 2010.

25 Leonid Libkin. Variable independence, quantifier elimination, and constraint representations. In *Proceedings of ICALP 2000*, volume 1853 of *LNCS*, pages 260–271. Springer, 2000. `doi:10.1007/3-540-45022-X\_23`.

26 Anthony W. Lin and Pablo Barceló. String solving with word equations and transducers: Towards a logic for analysing mutation XSS. In *Proceedings POPL 2016*, pages 123–136. ACM, 2016. `doi:10.1145/2837614.2837641`.

27 Christof Löding and Christopher Spinrath. Decision problems for subclasses of rational relations over finite and infinite words. In *Proceedings of FCT 2017*, volume 10472 of *LNCS*, pages 341–354. Springer, 2017. `doi:10.1007/978-3-662-55751-8\_27`.

28 Christof Löding and Christopher Spinrath. Decision problems for subclasses of rational relations over finite and infinite words. *Discrete Mathematics & Theoretical Computer Science*, 21(3), 2019. URL: `https://dmtcs.episciences.org/5141`.

29 Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory (SWAT 1972)*, pages 125–129. IEEE, 1972. `doi:10.1109/swat.1972.29`.

30 M. Nivat. Transduction des langages de Chomsky. *Ann. Inst. Fourier*, 18:339–455, 1968.

31 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

32 Michael Sipser. *Introduction to the Theory of Computation*. Thomson Course Technology Boston, second edition, 2006.

33 Richard Edwin Stearns. A regularity test for pushdown machines. *Information and Control*, 11(3):323–340, 1967. `doi:10.1016/S0019-9958(67)90591-8`.

34 A. W. To. *Model Checking Infinite-State Systems: Generic and Specific Approaches*. PhD thesis, LFCS, School of Informatics, University of Edinburgh, 2010.

35 A. W. To and Leonid Libkin. Recurrent reachability analysis in regular model checking. In *LPAR*, pages 198–213, 2008.

36 Minh-Thai Trinh, Duc-Hiep Chu, and Joxan Jaffar. S3: A symbolic string solver for vulnerability detection in web applications. In *Proceedings of CCS 2014*, pages 1232–1243. ACM, 2014. `doi:10.1145/2660267.2660372`.

37 Leslie G. Valiant. Regularity and related problems for deterministic pushdown automata. *Journal of the ACM*, 22(1):1–10, 1975. `doi:10.1145/321864.321865`.

38 Margus Veanes, Nikolaj Bjørner, Lev Nachmanson, and Sergey Bereg. Monadic decomposition. *Journal of the ACM*, 64(2):1–28, 2017. `doi:10.1145/3040488`.