Resynchronizing Classes of Word Relations

María Emilia Descotte
LaBRI, Université de Bordeaux

Diego Figueira
CNRS, LaBRI, Université de Bordeaux

Gabriele Puppis
CNRS, LaBRI, Université de Bordeaux

Abstract

A natural approach to define binary word relations over a finite alphabet $\mathbb{A}$ is through two-tape finite state automata that recognize regular languages over $\{1,2\} \times \mathbb{A}$, where (i,a) is interpreted as reading letter a from tape i. Accordingly, a word $w \in L$ denotes the pair $(u_1,u_2) \in \mathbb{A}^* \times \mathbb{A}^*$ in which u_i is the projection of w onto i-labelled letters. While this formalism defines the well-studied class of Rational relations (*a.k.a.* non-deterministic finite state transducers), enforcing restrictions on the reading regime from the tapes, which we call *synchronization*, yields various sub-classes of relations. Such synchronization restrictions are imposed through regular properties on the projection of the language onto $\{1,2\}$. In this way, for each regular language $C \subseteq \{1,2\}^*$, one obtains a class $\textsc{Rel}(C)$ of relations. Regular, Recognizable, and length-preserving rational relations are all examples of classes that can be defined in this way.

We study the problem of containment for synchronized classes of relations: given $C, D \subseteq \{1,2\}^*$, is $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$? We show a characterization in terms of C and D which gives a decidability procedure to test for class inclusion. This also yields a procedure to re-synchronize languages from $\{1,2\} \times \mathbb{A}$ preserving the denoted relation whenever the inclusion holds.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases synchronized word relations, containment, resynchronization

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.123

Funding Work supported by ANR project DELTA, grant ANR-16-CE40-0007, and LIA INFINIS.

1 Introduction

We study are relations of finite words, that is, binary relations $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$ for a finite alphabet $\mathbb{A}$. The study of these relations dates back to the works of Büchi, Elgot, Mezei, and Nivat in the 1960s [4, 8, 13], with much subsequent work done later (*e.g.*, [2, 6]). Most of the investigations focused on extending the standard notion of regularity from languages to relations. This effort has followed the long-standing tradition of using equational, operational, and descriptive formalisms – that is, finite monoids, automata, and regular expressions – for describing relations, and gave rise to three different classes of relations: the *Recognizable*, the *Automatic* (*a.k.a. Regular* [2] or *Synchronous* [6]), and the *Rational* relations.

The above classes of relations can be seen as three particular examples of a much larger (in fact infinite) range of possibilities, where relations are described by special languages over extended alphabets, called *synchronizing languages* [10]. Intuitively, the idea is to describe a binary relation by means of a two-tape automaton with two heads, one for each tape, which can move independently one of the other. In the basic framework of synchronized relations, one lets each head of the automaton to either move right or stay in the same

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 123; pp. 123:1–123:13


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

position. In addition, one can constrain the possible sequences of head motions by a suitable regular language $C \subseteq \{1,2\}^*$. In this way, each regular language $C \subseteq \{1,2\}^*$ induces a class of binary relations, denoted $\textsc{Rel}(C)$, which is contained in the class of Rational relations (due to Nivat's Theorem [13]). For example, the class of Recognizable, Automatic, and Rational relations are captured, respectively, by the languages $C_{\text{Rec}} = \{1\}^* \cdot \{2\}^*$, $C_{\text{Aut}} = \{12\}^* \cdot \{1\}^* \cup \{12\}^* \cdot \{2\}^*$, and $C_{\text{Rat}} = \{1,2\}^*$. However, it should be noted that other well-known subclasses of rational relations, such as deterministic or functional relations, are not captured by notion of synchronization. In general, the correspondence between a language $C \subseteq \{1,2\}^*$ and the induced class $\textsc{Rel}(C)$ of synchronized relations is not one-to-one: it may happen that different languages C, D induce the same class of synchronized relations. There are thus fundamental questions that arise naturally in this framework: *When do two classes of synchronized relations coincide, and when is one contained in the other?* Our contribution is a precise algorithmic answer to this type of questions.

More concretely, given a binary alphabet $\mathbb{2} = \{1,2\}$ and another finite alphabet $\mathbb{A}$, a word $w \in (\mathbb{2} \times \mathbb{A})^*$ is said to *synchronize* the pair $(w_1, w_2) \in \mathbb{A}^* \times \mathbb{A}^*$ if, for both $i = 1, 2$, w_i is the projection of w on $\mathbb{A}$ restricted to the positions marked with i. For short, we denote this by $[\![w]\!] = (w_1, w_2)$ – *e.g.*, $[\![(1,a)(1,b)(2,b)(1,a)(2,c)]\!] = (aba, bc)$. According to this definition, every word over $\mathbb{2} \times \mathbb{A}$ synchronizes a pair of words over $\mathbb{A}$, and every pair of words over $\mathbb{A}$ is synchronized by (perhaps many) words over of $\mathbb{2} \times \mathbb{A}$. This notion is readily lifted to languages: a language $L \subseteq (\mathbb{2} \times \mathbb{A})^*$ synchronizes the relation $[\![L]\!] = \{[\![w]\!] \mid w \in L\} \subseteq \mathbb{A}^* \times \mathbb{A}^*$. For example, $[\![((1,a)(2,a) \cup (1,b)(2,b))^*]\!]$ denotes the equality relation over $\mathbb{A} = \{a,b\}$.

In this setup, one can define classes of relations by restricting the set of admitted synchronizations. The natural way of doing so is to fix a language $C \subseteq \mathbb{2}^*$, called *control language*, and let L vary over all regular languages over the alphabet $\mathbb{2} \times \mathbb{A}$ whose projections onto $\mathbb{2}$ are in C. Thus, for every regular $C \subseteq \mathbb{2}^*$, there is an associated class $\textsc{Rel}(C)$ of C-controlled relations, namely, relations synchronized by regular languages $L \subseteq (\mathbb{2} \times \mathbb{A})^*$ whose projection onto $\mathbb{2}$ are in C. Clearly, $C \subseteq D \subseteq \mathbb{2}^*$, implies $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$, but the converse does not hold: while $\textsc{Rel}(C_{\text{Rec}}) = \textit{Recognizable} \subseteq \textit{Automatic} = \textsc{Rel}(C_{\text{Aut}})$, we have $C_{\text{Rec}} \not\subseteq C_{\text{Aut}}$. Moreover, as we have mentioned earlier, different control languages may induce the same class of synchronized relations. For example, once again, the class of Recognizable relations is induced by the control language $C_{\text{Rec}} = \{1\}^*\{2\}^*$, but also by $C'_{\text{Rec}} = \{1\}^*\{2\}^*\{1\}^*$, and the class of Automatic relations is induced by $C_{\text{Aut}} = \{12\}^* \cdot \{1\}^* \cup \{12\}^* \cdot \{2\}^*$, or equally by $C'_{\text{Aut}} = \{21\}^* \cdot \{1\}^* \cdot \{2\}^*$. This 'mismatch' between control languages and induced classes of relations gives rise to the following algorithmic problem.

Class Containment Problem	
Input:	Two regular languages $C, D \subseteq \mathbb{2}^*$
Question:	Is $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$?

Note that the above problem is different from the (C, D)-membership problem on synchronized relations, which consists in deciding whether $R \in \textsc{Rel}(D)$ for a given $R \in \textsc{Rel}(C)$, and which can be decidable or undecidable depending on C, D [5]. The Class Containment Problem can be seen as the problem of whether *every* C-controlled regular language L has a D-controlled regular language L' so that $[\![L]\!] = [\![L']\!]$. It was proved in [10] that this problem is decidable for some particular instances of D, namely, for $D =$ *Recognizable*, *Automatic*, *Length-preserving* or *Rational*. More specifically, given a regular language C over the binary alphabet $\mathbb{2}$, it is decidable whether $\textsc{Rel}(C)$ is contained or not in *Recognizable* (respectively, *Automatic*, *Length-preserving* and *Rational*). Our main contribution is a procedure for deciding the Class Containment Problem in full generality, i.e. for arbitrary C and D.

▶ **Main Theorem.** *The Class Containment Problem is decidable.*

In addition, our results show that, for positive instances (C, D), one can effectively transform any regular C-controlled language L into a regular D-controlled language L' so that $[\![L]\!] = [\![L']\!]$. By 'effectively transform' we mean that one can receive as input an automaton (or a regular expression) for L and produce an automaton (or a regular expression) for L'. In particular, we show a normal form of control languages, implying that every synchronized class can be expressed through a control language of star-height at most 1.

Related work. The formalization of a framework in which one can describe classes of word relations by means of synchronization languages is quite recent [10]. As already mentioned, the class containment problem was only addressed for the classes of Recognizable, Automatic and Rational relations, for which several characterizations have been proposed [10]. The formalism of synchronizations has been extended beyond rational relations by means of semi-linear constraints [9] in the context of path querying languages for graph databases.

The paper [3] studies relations with origin information, as induced by non-deterministic (one-way) finite state transducers. Origin information can be seen as a way to describe a synchronization between input and output words – somehow in the same spirit of our synchronization languages – and was exploited to recover decidability of the equivalence problem for transducers. The paper [11] pursues further this principle by studying "distortions" of the origin information, called resynchronizations. Despite the similar terminology and the connection between origins and synchronizing languages, the problems studied in [3, 11] are of rather different nature than our Class Containment Problem.

Organization. After the preliminaries on subclasses of regular languages, we define in Section 3 the framework of synchronized relations. Section 4 provides a roadmap with the three key ingredients of our characterization. Sections 5, 6 and 7 contain the technical details for these main ingredients. In Section 8 we discuss the computability of the characterization.

2 Preliminaries

We denote by $\mathbb{N}, \mathbb{Q}$ the sets of non-negative integers and rationals. We use standard interval notation as in, for example, $(a, b]_{\mathbb{Q}} = \{c \in \mathbb{Q} \mid a < c \leq b\}$. $\mathbb{A}, \mathbb{B}$ denote arbitrary finite alphabets, and $\mathbb{2}$ the special binary alphabet $\{1, 2\}$.

Words and shuffles. For a word $w \in \mathbb{A}^*$, $|w|$ is its length, and $|w|_a$ is the number of occurrences of symbol a in w. We denote by $w[i, j]$ the factor of w between positions i and j (included), for $1 \leq i \leq j \leq |w|$, and we write $w[i]$ for $w[i, i]$. We will also make use of the **shuffle** operation, which maps a finite set of words $w_1, \ldots, w_n$ to the language $\mathrm{shuffle}\{w_1, \ldots, w_n\}$ of all words w for which there is a partition $I_1, \ldots, I_n$ of $[1, |w|]$ so that each w_i is the projection of w onto I_i. For example, $\mathrm{shuffle}\{ab, cd\} = \{abcd, cdab, acbd, acdb, \ldots\}$.

Parikh image. The **Parikh image** of a word w over $\mathbb{A}$ is the tuple $\pi(w)$ associating each symbol $a \in \mathbb{A}$ to its number of occurrences $|w|_a$ in w. We will mostly use Parikh images for words over $\mathbb{2}^*$, which are thus pairs $\pi(w) = (|w|_1, |w|_2)$. We naturally extend this to languages by letting $\pi(L) \stackrel{\text{def}}{=} \{\pi(w) \mid w \in L\}$ $(\subseteq \mathbb{N}^2)$. For $\bar{x}, \bar{x}_1, \ldots, \bar{x}_n \in \mathbb{N}^2$, we denote by $\langle \bar{x}, P \rangle$ the 2-dimensional linear set $\{\bar{x} + \alpha_1 \bar{x}_1 + \cdots + \alpha_n \bar{x}_n \mid \alpha_1, \ldots, \alpha_n \in \mathbb{N}\}$, and call $\bar{x} \in \mathbb{N}^2$ its **basis** and $\bar{x}_1, \ldots, \bar{x}_n$ its **periods**. A **semi-linear** set is a finite union of linear sets.

Regular languages. We use standard notation for regular expressions without complement, namely, for expressions build up from the empty set, the empty word ε and the symbols $a \in \mathbb{A}$, using the operations $\cdot$, $\cup$, and $(\,)^*$. For economy of space and clarity we also use the abbreviated notation $(\,)^k$, $(\,)^{k*}$ – which is a shorthand for $((\,)^k)^*$ – , $(\,)^{\geq k}$, $(\,)^{<k}$, and we identify regular expressions with the defined languages. For example, we may write $abbc \in a \cdot b^{\geq 2} \cdot (c \cup d)^*$, $b(ab)^* = (ba)^*b$ and $\{a, b\}^* \cdot c = (a \cup b)^* \cdot c$. Given $u = a_1 \cdots a_n \in \mathbb{A}^*$ and $v = b_1 \cdots b_n \in \mathbb{B}^*$, we write $u \otimes v$ for the word $(a_1, b_1) \cdots (a_n, b_n) \in (\mathbb{A} \times \mathbb{B})^*$. Similarly, given $U \subseteq \mathbb{A}^*$, $V \subseteq \mathbb{B}^*$, we write $U \otimes V \subseteq (\mathbb{A} \times \mathbb{B})^*$ for the set $\{u \otimes v \mid u \in U, v \in V, |u| = |v|\}$.

The **star-height** of a regular expression is the maximum number of nestings of Kleene stars $(\,)^*$. By abuse of terminology, when referring to the star-height of a language, we mean the star-height of some regular expression that represents it (in particular, we do not need to work with the minimum star-height over all expressions). Besides regular expressions, we also work with automata, and use classical techniques on them (notably, pumping arguments). Given an accepting run γ of an automaton $\mathcal{A}$, we often identify **cycles** in it, that is, factors that start and end in the same state, and that can thus be pumped. Such cycles are called **simple** if they do not contain proper factors that are also cycles. Moreover, to avoid mentioning explicitly an automaton for a language L and a run of it, we call **cycle of** L (resp. **simple cycle of** L) the word spelled out by any cycle (resp. simple cycle) of any accepting run of the minimal deterministic automaton recognizing L, and denote the set of all cycles (resp. simple cycles) of L by *cycles*(L) (resp. *simple-cycles*(L)). We remark that, however, that the use of the minimal automaton as a presentation of a regular language L is only to avoid ambiguity when referring to the cycles of L – in fact, our results do not depend on determinism or minimality, and can thus be applied to arbitrary non-deterministic automata, without any difference in the characterizations we present.

A regular language C is **concat-star** (*a.k.a. unit-form* [1]), if it is of the form

$$C = C_1^* u_1 C_2^* u_2 \cdots C_n^* u_n, \tag{$\star$}$$

for $n \in \mathbb{N}$, words $u_1, \ldots, u_n$, and regular languages $C_1, \ldots, C_n$. Without loss of generality, we can always assume that the empty word does not belong to any of the languages C_i. The following trivial decomposition lemma will be used throughout.

▶ **Lemma 1.** *Every regular language is a finite union of concat-star languages.*

The C_i^*'s from $(\star)$ are called **components** of the concat-star language C. Note that (an expression of) a concat-star language as in $(\star)$ has star-height 1 if and only if every C_i is finite. A component C_i^* is **homogeneous** if $C_i^* \subseteq 1^*$ or $C_i^* \subseteq 2^*$. A component which is not homogeneous is called **heterogeneous** (e.g. $C_i^* = \{1, 2\}^*$). It will also be convenient to distinguish a few types of concat-star languages. We say that C is

- **heterogeneous** if it contains at least one heterogeneous component, otherwise it is **homogeneous**;
- **smooth** if every homogeneous component is a language of the form 1^{k*} or 2^{k*}, for some $k > 0$, and there are no consecutive homogeneous components;
- **simple** if it has star-height 1 and it is either homogeneous or smooth heterogeneous.

Hereafter, by "simple language" we mean simple concat-star language. The picture below summarizes the different types of control languages, together with some separating examples.

	homogeneous	smooth heterogeneous	non-smooth heterogeneous	non concat-star
s.-h. > 1	$(1^*1)^*2^*$	$1^*(1^*2)^*2^*$	$1^*2^*(1^*2)^*$	$(1^*2)^* \cup (12)^*$
s.-h. $= 1$	$1^*(11)^*2^*$	$1^*(12)^*2^*$	$1^*2^*(12)^*$	$(12)^*1^* \cup (12)^*2^*$
	simple			

In Section 5 we will see that the Class Containment Problem is reduced to the case of finite unions of simple languages. The latter languages thus form the basis of our characterization.

3 Synchronized relations

A **synchronization** of a pair (w_1, w_2) of words over $\mathbb{A}$ is a word over $\mathbb{2} \times \mathbb{A}$ so that the projection on $\mathbb{A}$ of positions labeled i is exactly w_i, for $i = 1, 2$ – in other words, $\text{shuffle}\{1^{|w_1|} \otimes w_1, 2^{|w_2|} \otimes w_2\}$ is the set of all synchronizations of (w_1, w_2). For example, the words $(1, a)(1, b)(2, a)$ and $(1, a)(2, a)(1, b)$ are two possible synchronizations of the same pair (ab, a). Every word $w \in (\mathbb{2} \times \mathbb{A})^*$ is a synchronization of a unique pair (w_1, w_2), where w_i is the sequence of $\mathbb{A}$-letters corresponding to the symbol i in the first position of $\mathbb{2} \times \mathbb{A}$. We denote such pair (w_1, w_2) by $[\![w]\!]$ and extend the notation to languages $L \subseteq (\mathbb{2} \times \mathbb{A})^*$ by $[\![L]\!] \stackrel{\text{def}}{=} \{[\![w]\!] \mid w \in L\}$.

Given a regular language $C \subseteq \mathbb{2}^*$, we define the class of ***C*-controlled relations** as

$$\textsc{Rel}(C) \stackrel{\text{def}}{=} \big\{[\![L]\!] \;\mid\; L \subseteq C \otimes \mathbb{A}^* \text{ is regular}, \mathbb{A} \text{ is some finite alphabet}\big\}.$$

A slightly different definition is possible, which restricts the class of C-controlled relations to be over a fixed alphabet $\mathbb{A}$, that is, one can define $\textsc{Rel}_{\mathbb{A}}(C) = \{[\![L]\!] \mid L \subseteq C \otimes \mathbb{A}^* \text{ regular}\}$. As far as we are concerned with comparing classes of relations controlled by different languages, the two definitions are somehow interchangeable, in the sense that containment between classes is not sensible to whether we fix or not the alphabet. For example, we will see that, for any alphabet $\mathbb{A}$ with at least two symbols, $\textsc{Rel}_{\mathbb{A}}(C) \subseteq \textsc{Rel}_{\mathbb{A}}(D)$ iff $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$.

For economy of space, we use $C \subseteq_{\textsc{Rel}} D$ and $C =_{\textsc{Rel}} D$ as shorthands for $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$ and $\textsc{Rel}(C) = \textsc{Rel}(D)$, respectively. The following properties are easy to verify.

▶ **Lemma 2.** *For every regular $C, D, C', D' \subseteq \mathbb{2}^*$,*

P1. *if $C \subseteq D$, then $C \subseteq_{\textsc{Rel}} D$;*
P2. *if $C \subseteq_{\textsc{Rel}} D$ and $C' \subseteq_{\textsc{Rel}} D'$, then $C \cdot C' \subseteq_{\textsc{Rel}} D \cdot D'$ and $C \cup C' \subseteq_{\textsc{Rel}} D \cup D'$;*
P3. *if $C \subseteq_{\textsc{Rel}} D$, then $C^* \subseteq_{\textsc{Rel}} D^*$;*
P4. *if $C \subseteq 1^*$ and $D \subseteq 2^*$, then $C \cdot D =_{\textsc{Rel}} D \cdot C$;*
P5. *if C is finite, then $C \cdot D =_{\textsc{Rel}} D \cdot C$;*
P6. *if $C \subseteq_{\textsc{Rel}} D$ then $\pi(C) \subseteq \pi(D)$; moreover, if C is finite, the converse also holds;*
P7. *if C is homogeneous concat-star, then $C =_{\textsc{Rel}} \bigcup_{i \in I} 1^{\ell_i *} 1^{k_i} 2^{\hat{\ell}_i *} 2^{\hat{k}_i}$ for a finite I;*
P8. *if C is homogeneous concat-star, $C \subseteq_{\textsc{Rel}} D$ if and only if $\pi(C) \subseteq \pi(D)$.*

Proof idea. P1 is immediate from definitions; henceforth we use it without referencing it. P2 and P3 follow readily from the following decomposition properties.

(a) For every $R \in \textsc{Rel}(C \cdot C')$, there are $R_1, \ldots, R_n \in \textsc{Rel}(C)$, $R'_1, \ldots, R'_n \in \textsc{Rel}(C')$ so that $R = \bigcup_i R_i \cdot R'_i$.
(b) For every $R \in \textsc{Rel}(C \cup C')$, there are $R_1 \in \textsc{Rel}(C)$, $R_2 \in \textsc{Rel}(C')$ so that $R = R_1 \cup R_2$.

(c) For every $R \in \text{REL}(C^*)$, there are $R_1, \ldots, R_n \in \text{REL}(C)$ and $I \subseteq \{1, \ldots, n\}^*$ regular so that $R = \bigcup_{w \in I} R_{w[1]} \cdots R_{w[|w|]}$.

P4 can be verified by first decomposing any relation $R \in \text{REL}(C \cdot D)$ into $\bigcup_i R_i \cdot R'_i$ as in (a), and then observing that in this case $[\![\bigcup_i R_i \cdot R'_i]\!] = [\![\bigcup_i R'_i \cdot R_i]\!]$. For P5, it is easy to see that $1 \cdot D =_{\text{REL}} D \cdot 1$ and $2 \cdot D =_{\text{REL}} D \cdot 2$ for any D, and thus by P2 this extends to commuting with arbitrary finite languages. For P6, observe that if $C \subseteq_{\text{REL}} D$ then $[\![C \otimes a^*]\!] \in \text{REL}(D)$ for $a \in \mathbb{A}$, which means that $\pi(C) \subseteq \pi(D)$. P7 is a consequence of P4 and the so-called Chrobak normal form for regular languages over unary alphabets [7]. Finally, the proof of P8 is a variant of the proof that the operation of shuffle preserves regularity of languages. ◀

4 Characterization of the Class Containment Problem

We give an overview of the main ingredients of our decision procedure for class containment.

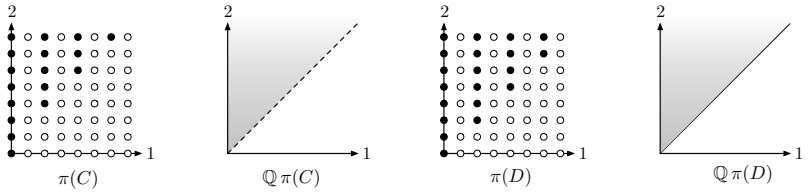
Decomposition. A first ingredient is a decomposition result for regular control languages into $=_{\text{REL}}$-equivalent finite unions of simple languages. Here we only state the result with a short proof sketch; the complete proof will be given in Section 5.

▶ **Proposition 3.** *Every regular language $C \subseteq 2^*$ is effectively $=_{\text{REL}}$-equivalent to a finite union of simple languages.*

Proof idea. One first applies Lemma 1, so as to decompose the regular language C into a finite union of concat-star languages. Then, the concat-star languages are further decomposed into unions of concat-star languages of star-height 1. For example, $(112(12)^* \cup 122)^* =_{\text{REL}} (122)^* \cup (112 \cup 122)^*112 \cup (112 \cup 122)^*11122 \cup (112 \cup 122)^*1111222$. This latter step is more difficult and exploits the increased flexibility of the relation $=_{\text{REL}}$ compared to equality. It also exploits in a crucial way properties of linear sets, and more specifically those that result from taking the Parikh images of concat-star languages. Finally, to get the desired decomposition, one needs to decompose further the concat-star languages of star-height 1 into finite unions of simple languages as in, for example, $(12)^*1^*2^* =_{\text{REL}} (12)^*1^* \cup (12)^*2^*$. This last decomposition makes use of some basic properties from Lemma 2. ◀

Parikh ratios. The **Parikh ratio** of a pair $\bar{x} = (n_1, n_2) \in \mathbb{N}^2 \setminus \{(0,0)\}$ is $\rho(\bar{x}) = \frac{n_1}{n_1+n_2}$. We naturally extend this to non-empty words $w \in 2^*$ by letting $\rho(w) = \rho(\pi(w))$ (this describes the proportion of 1's in w). We further extend the notation to languages: $\rho(C) = \{\rho(w) \mid w \in C \setminus \{\varepsilon\}\}$. Note that $\rho(C) \subseteq [0,1]_{\mathbb{Q}}$. It is sometimes useful to think of $\rho(C)$ as the cone $\mathbb{Q}\,\pi(C) = \{q \cdot \pi(w) \mid q \in \mathbb{Q}, w \in C\}$ inside the rational plane $\mathbb{Q} \times \mathbb{Q}$.

▶ **Example 4.** The Parikh images of the languages $C = (2(2112)^*)^*$ and $D = (2 \cup 2112)^*$ are depicted below. Note that $\rho(C) = \left[0, \frac{1}{2}\right)_{\mathbb{Q}}$, while $\rho(D) = \left[0, \frac{1}{2}\right]_{\mathbb{Q}}$.



The following lemma summarizes the main properties of Parikh ratios that we will need.

▶ **Lemma 5.** *The Parikh ratio of a concat-star language C verifies the following properties:*

1. *If $C = C_1^* u_1 \cdots C_n^* u_n$, then $\rho(\text{cycles}(C)) \subseteq [\min_i \inf \rho(C_i^*), \max_i \sup \rho(C_i^*)]_{\mathbb{Q}}$;*
2. *Moreover, if $C = D^*$ for a finite D, then $\rho(C) = \rho(\text{cycles}(C)) = [\min \rho(D), \max \rho(D)]_{\mathbb{Q}}$.*

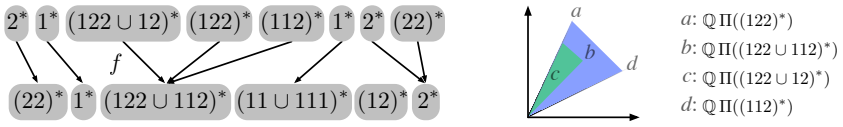
Synchronizing morphisms. Another fundamental ingredient is the notion of *synchronizing morphism*, which intuitively relates the components of a concat-star language C to the components of a concat-star language D by comparing the Parikh ratios.

Let $C = C_1^* u_1 \cdots C_n^* u_n$ be a heterogeneous concat-star language and $D = D_1^* v_1 \cdots D_m^* v_m$ any concat-star language. We say that a function $f : [1, n] \to [1, m]$ is a **synchronizing morphism** (abbreviated *s.m.*) from C to D if

- it is monotonic: $f(i) \leq f(j)$ whenever $i \leq j$; and
- it preserves Parikh-ratio: for every $i \in [1, n]$, $\rho(C_i^*) \subseteq \rho(D_{f(i)}^*)$.

We write $C \xrightarrow{s.m.} D$ to denote the existence of such synchronizing morphism. By convention, if C is homogeneous, then we say that there is always a synchronizing morphism from C to D. In particular, $u \xrightarrow{s.m.} v$ for every $u, v \in \mathbb{2}^*$. The sole purpose of this trivial definition on homogeneous concat-star languages is to make the characterization statements simpler.

▶ **Example 6.** The following function f is a synchronizing morphism:



Observe that synchronizing morphisms are closed under composition and hence $\xrightarrow{s.m.}$ defines a pre-order on concat-star languages.

Class Containment Problem for simple languages. The existence of synchronizing morphism is the key property that characterizes $\subseteq_{\text{Rel}}$ on simple languages. A complete proof of the following proposition will be the theme of Section 6.

▶ **Proposition 7.** *For all simple $C, D \subseteq \mathbb{2}^*$, $C \subseteq_{\text{Rel}} D$ iff $\pi(C) \subseteq \pi(D)$ and $C \xrightarrow{s.m.} D$.*

Note that the case of C homogeneous follows from P8. Intuitively, for any C smooth heterogeneous concat-star language of star-height 1, the characterization says that, $C \subseteq_{\text{Rel}} D$ iff $\pi(C) \subseteq \pi(D)$ and for every component of C, there is a component of D that contains its Parikh ratio. Further, the matching between components is monotonic. For example, we have $(12)^*(112)^* \subseteq_{\text{Rel}} (12 \cup 11122)^*(121)^*1^*2^*$, because the Parikh ratios of $(12)^*$ and $(112)^*$ are included in those of $(12 \cup 11122)^*$ and $(121)^*$, respectively. On the other hand, we have $(112)^*(12)^* \not\subseteq_{\text{Rel}} (12 \cup 11122)^*(121)^*1^*2^*$ because in this case there is no motononic matching between components.

Generalization to unions of simple languages. Section 7 concerns the generalization of the characterization to finite unions of simple languages, which cover arbitrary regular languages up to $=_{\text{Rel}}$-equivalence. The previous characterization for simple languages thus constitutes the base case of our characterization. The lemma below allows a first generalization when C is a union of simple languages and D is a simple language.

▶ **Lemma 8.** *$C_1 \cup C_2 \subseteq_{\text{Rel}} D$ iff $C_1 \subseteq_{\text{Rel}} D$ and $C_2 \subseteq_{\text{Rel}} D$.*

The analogous of Lemma 8 for unions on the right hand-side does not hold in general, as shown by the following example.

▶ **Example 9.** Let $C = (12)^*$, $D_1 = (112 \cup 1122)^*$, and $D_2 = (122 \cup 1122)^*12$. We have $C \subseteq_{\text{Rel}} D_1 \cup D_2$, although $C \not\subseteq_{\text{Rel}} D_1$ and $C \not\subseteq_{\text{Rel}} D_2$.

Neither it holds that Parikh image containment together with the existence of s.m. to one of the disjuncts suffices. For instance, for $C' = (12)^*$, $D'_1 = (1212)^*$, $D'_2 = 1^*2^*$, we have $C' \not\subseteq_{\text{Rel}} D'_1 \cup D'_2$ although $\pi(C') \subseteq \pi(D'_1 \cup D'_2)$ and $C' \xrightarrow{s.m.} D'_1$.

The characterization we provide is inductive on the number of languages that are unioned on the right hand-side. Concretely, for a union of two languages, we will show that $C \subseteq_{\text{Rel}} D_1 \cup D_2$ iff $C \xrightarrow{s.m.} D_i$ for some i and $C \setminus [D_i]_\pi \subseteq_{\text{Rel}} D_{3-i}$, where $[D_i]_\pi$ is the closure of D_i under permutations, that is, $[D_i]_\pi \stackrel{\text{def}}{=} \{w \in \mathbb{2}^* \mid \pi(w) \in \pi(D_i)\}$. The idea that underlies the proof of the necessity of our characterization is that C can be split into a disjoint union of $C \cap [D_i]_\pi$ and $C \setminus [D_i]_\pi$, in such a way that $C \cap [D_i]_\pi \subseteq_{\text{Rel}} D_i$ and $C \setminus [D_i]_\pi \subseteq_{\text{Rel}} D_{3-i}$.

For finite unions of simple languages, we have the following characterization. A complete proof of this theorem will be the theme of Section 7.

▶ **Theorem 10.** *For finite unions $C = \bigcup_i C_i$ and $D = \bigcup_j D_j$ of simple languages, the following are equivalent:*

- $C \subseteq_{\text{Rel}} D$,
- *For all i $\pi(C_i) \subseteq \pi(D)$ and there is j with $C_i \xrightarrow{s.m.} D_j$. In addition, if C_i is heterogeneous, then $C_i \setminus [D_j]_\pi$ is regular and $C_i \setminus [D_j]_\pi \subseteq_{\text{Rel}} \bigcup_{j' \neq j} D_{j'}$.*

Coming back to Example 9, note that $\rho(C) = \{\frac{1}{2}\}$, $\rho(D_1) = [\frac{1}{2}, \frac{2}{3}]_{\mathbb{Q}}$ and $\rho((122 \cup 1122)^*) = [\frac{1}{3}, \frac{1}{2}]_{\mathbb{Q}}$. Therefore, one can explain $C \subseteq_{\text{Rel}} D_1 \cup D_2$ by the fact of having $C \xrightarrow{s.m.} D_1$ and $C \setminus [D_1]_\pi = (1212)^*12 \subseteq_{\text{Rel}} D_2$, where the latter containment holds by the fact that $(1212)^*12 \xrightarrow{s.m.} D_2$ and $\pi((1212)^*12) \subseteq \pi(D_2)$.

Note that there's a caveat in the statement of Theorem 10: $C_i \setminus [D_j]_\pi$ needs to be *regular*. And in fact this is not the case in general: if $C_i = 1^*2^*$ and $D_j = (12)^*$, we get a non-regular language $C_i \setminus [D_j]_\pi = \{1^n 2^m \mid n \neq m\}$. However, provided $C_i \xrightarrow{s.m.} D_j$ for C_i heterogeneous, we show that $C_i \setminus [D_j]_\pi$ is effectively regular (in the sense that an automaton recognizing it can be computed from automata recognizing C_i and D_j). This is a non-trivial fact, and will be proved in Section 5 (Proposition 12).

The second key ingredient is that if $C_i \subseteq_{\text{Rel}} D_1 \cup \cdots \cup D_n$, then there must be some j so that $C_i \xrightarrow{s.m.} D_j$. This will be proved in Section 6 (Lemma 15).

5 Decomposition into simple languages

As already mentioned, we start by reducing the Class Containment Problem for arbitrary regular languages to the case of finite unions of simple languages (Proposition 3 below). We do this in two steps. First, we decompose regular languages into finite unions of concat-star languages of star-height 1 (Lemma 13 below). Then, we further decompose the latter languages into finite unions of simple languages (Lemma 14 below).

Unions of star-height 1 languages. Lemma 13 relies on two key results, which are also of independent interest. The first result is a normal form representation of the Parikh image $\pi(C)$ of a concat-star language C. Formally, we say that a linear set $\langle \bar{x}, P \rangle$ is in **normal form** if the elements of P are linearly independent. We extend this notion to semi-linear sets by saying that $\langle \bar{x}_1, P_1 \rangle \cup \cdots \cup \langle \bar{x}_n, P_n \rangle$ is in normal form if the vectors in $\bigcup_i P_i$ are linearly independent. In particular, in dimension 2, this means that there are at most two vectors in P. Note that if the representation of a semi-linear set is in normal form then all its linear sets are in normal form, but the converse does not hold – for example, consider $\langle \bar{0}, \{(2,0)\} \rangle \cup \langle \bar{0}, \{(3,0)\} \rangle$. The following lemma shows that Parikh images of concat-star languages enjoy normal forms.

▶ **Lemma 11.** *For every concat-star language $C = C_1^* u_1 \cdots C_n^* u_n$, there exists a normal form representation of its Parikh image $\pi(C)$. Moreover, if C is infinite, the union of the period sets is $\{\bar{x}_-, \bar{x}_+\}$, where $\rho(\bar{x}_-) = \min_j(\inf \rho(C_j^*))$ and $\rho(\bar{x}_+) = \max_j(\sup \rho(C_j^*))$.*

Proof idea. Using some basic properties of Parikh images, we reduce to the case where C is a concatenation of expressions of the form u^* (for u a non-empty word) or $(u_1^* \cdots u_n^* u)^*$ (for $u_1, \ldots, u_n, u$ non-empty words). For any C in this form, the Parikh image of words in C can be expressed in terms of some words w_-, w_+ such that $\pi(w_-) = \bar{x}_-$ and $\pi(w_+) = \bar{x}_+$. Then, any word of C can be represented as a constrained iteration of these two words. ◀

It is worth pointing out the difference with the normal form from [12]. While the normal form of [12] holds for arbitrary regular languages, our normal form holds only for *concat-star* languages over *binary* alphabets (*e.g.*, it fails for $(12)^* \cup 1^* \cup 2^*$). Conversely, the normal form from [12] does not guarantee the linear independence of the vectors in the union of the periods, as we do here instead. Proposition 12 below relies on such an additional property. (Also, [1] gives a procedure to compute Parikh images, though no normal form is implied.)

The second result shows that, under certain conditions, one can intersect a regular language C by a language of the form $[D]_\pi = \pi^{-1}(\pi(D))$, with D concat-star, and obtain a language that is again regular. This result not only enables the decomposition into star-height 1 languages, but will be used also later to formalize a recursive characterization of $\subseteq_{\text{REL}}$ for unions of simple languages (cf. Section 7).

▶ **Proposition 12.** *Given C regular and D concat-star so that $\rho(\mathit{cycles}(C)) \subseteq \rho(\mathit{cycles}(D))$, the languages $C \cap [D]_\pi$ and $C \setminus [D]_\pi$ are effectively regular. If in addition D is of the form $D_1^* u$, then $C \cap [D]_\pi \subseteq_{\text{REL}} D$.*

Proof idea. We exploit the fact that words in $\mathbb{2}^*$ are in bijection with paths inside $\mathbb{N}^2$ that originate in $\bar{0} = (0, 0)$ and, furthermore, that words with the same Parikh image correspond to paths with the same endpoints. The claim boils down to considering some word $w \in \mathbb{2}^*$ and proving that, under suitable hypotheses, the path induced by w can be approximated by a path inside $\pi(D)$ that stays sufficiently close to the former path. The use of Lemma 11 will be crucial here, since it gives a normal form $\bigcup_i \langle \bar{x}_i, P_i \rangle$ for the latter set $\pi(D)$. Intuitively, it implies that the words from $[D]_\pi$ are represented by paths that never get too far from the linear set $\langle \bar{0}, \bigcup_i P_i \rangle$. For example, by pairing this property with the assumption that $\rho(\mathit{cycles}(C)) \subseteq \rho(\mathit{cycles}(D))$, one can show that the path induced by a word $w \in C$ stays close to $\langle \bar{0}, \bigcup_i P_i \rangle$, and hence also to $\pi(D)$. Stronger variants of this property are shown, that take into account the exact displacement of points along the path induced by w from the points in $\pi(D)$. These latter properties are used by suitable automata that recognize the languages $C \cap [D]_\pi$ and $C \setminus [D]_\pi$. ◀

As we explained in the proof sketch, the above proposition relies on the normal form for the semi-linear set $\pi(D)$, which in turns relies on the fact that D is concat-star. The proposition does not hold if we replace D with an arbitrary regular language. For instance, consider $C = 1(11)^*2(22)^*$ and $D = (12)^* \cup (11)^*(22)^*$, and observe that $\rho(\mathit{cycles}(C)) = [0, 1]_{\mathbb{Q}} = \rho(\mathit{cycles}(D))$, but $C \cap [D]_\pi = \{1(11)^n 2(22)^n \mid n \in \mathbb{N}\}$ is clearly not regular.

Although Proposition 12 is stated in full generality, that is, for every regular language C so that $\rho(\mathit{cycles}(C)) \subseteq \rho(\mathit{cycles}(D))$, in the proof of the decomposition result below we will use it only for a smooth heterogeneous concat-star language C so that $C \xrightarrow{s.m.} D$ (this is sufficient but not necessary for verifying the hypothesis $\rho(\mathit{cycles}(C)) \subseteq \rho(\mathit{cycles}(D))$).

▶ **Lemma 13.** *Every regular $C \subseteq \mathbb{2}^*$ is $=_{\text{REL}}$-equivalent to a finite union $\bigcup_i D_i$ of concat-star languages of star-height* 1.

Towards the proof of this lemma, note that, by Lemma 1, C is a finite union of concat-star languages $C_1^* u_1 \cdots C_n^* u_n$. The lemma then follows from applying Claim 1 below to each component of the concat-star languages, and then using P2.

▶ **Claim 1.** *Every regular D^* is $=_{\text{REL}}$-equivalent to a finite union $\bigcup_i D_i^* u_i$, with finite D_i's.*

Proof idea of Claim 1. Since $\pi(D^*)$ is a finite union of linear sets, from the latter we can extract languages of the form $D_i^* u_i$. Then we can decompose D^* as the union of $D^* \cap [D_i^* u_i]_\pi$. From there, the result follows easily from Proposition 12 and P2. ◀

Unions of simple languages. We finally show how to decompose into simple languages.

▶ **Lemma 14.** *Every concat-star $C \subseteq \mathbb{2}^*$ of star-height 1 is $=_{\text{REL}}$-equivalent to a finite union $\bigcup_i C_i$ of simple languages.*

Proof idea. By using the basic properties given in Lemma 2, we can reduce the problem to the case where C is of the form $1^{k*}\hat{2}^{\hat{k}*}w^*$ for some heterogeneous word w and some natural numbers $k, \hat{k}$. This case is easy to prove by using again those basic properties. ◀

As a corollary of Lemmas 13 and 14, we have our desired result.

▶ **Proposition 3.** Every regular language $C \subseteq \mathbb{2}^*$ is effectively $=_{\text{REL}}$-equivalent to a finite union of simple languages.

6 Simple languages

We prove the characterization result for simple languages, which we recall here.

▶ **Proposition 7.** For all simple $C, D \subseteq \mathbb{2}^*$, $C \subseteq_{\text{REL}} D$ iff $\pi(C) \subseteq \pi(D)$ and $C \xrightarrow{s.m.} D$.

For the left-to-right direction, by P6, $C \subseteq_{\text{REL}} D$ implies $\pi(C) \subseteq \pi(D)$. The proof that $C \subseteq_{\text{REL}} D$ implies $C \xrightarrow{s.m.} D$ is given in a more general setup where D is a finite union of simple languages. This statement will be used in the characterization of the next section.

▶ **Lemma 15.** *For C a simple language and $D = \bigcup_i D_i$ finite union of simple languages, if $C \subseteq_{\text{REL}} D$, then $C \xrightarrow{s.m.} D_i$ for some i. In particular, for C, D simple languages, if $C \subseteq_{\text{REL}} D$, then $C \xrightarrow{s.m.} D$. Further, the statement holds even if we consider $\text{REL}_{\mathbb{A}}$-containment for any $\mathbb{A}$ with at least two letters.*

Proof idea. The idea is to construct a relation $R \in \text{REL}(C)$ so that from $R \in \text{REL}(D)$, using suitable pumping arguments, one can extract a synchronizing morphism from C to some D_i. The relation R must depend on both languages C, D, but the underlying alphabet can be fixed and taken binary, say $\mathbb{A} = \{a, b\}$. For example, if C is of the form C_1^* and contains two words u^- and u^+ with minimum and maximum Parikh ratios, and if the automaton for D has a single strongly connected component, then one can define the relation $R = [\![(u^- \otimes a^{|u^-|})^* \cdot (u^+ \otimes b^{|u^+|})^*]\!]$. In this case, $R \in \text{REL}(D)$ would imply $\rho(u^-), \rho(u^+) \in \rho(D)$, and hence $C \xrightarrow{s.m.} D$. This construction can be modified for more general languages C, D, by using words with different Parikh ratios from each component of C and by increasing the number of alternations between these ratios on the basis of the number of components of D. While the construction is more involved in the general case, and in particular needs to include iterations of words which are not necessarily of minimum or maximum Parikh ratios for a component, the intuition remains the same. ◀

▶ **Observation 16.** *The previous Lemma 15 does not hold for arbitrary concat-star languages C. For example, consider $(12)^*1^*2^* =_{\text{Rel}} (12)^*1^* \cup (12)^*2^*$, where there is no s.m. from $(12)^*1^*2^*$ to $(12)^*1^*$, nor from $(12)^*1^*2^*$ to $(12)^*2^*$.*

Conversely, to show that the conditions $\pi(C) \subseteq \pi(D)$ and $C \xrightarrow{s.m.} D$ are sufficient to have $C \subseteq_{\text{Rel}} D$, where C, D are simple, it is useful to introduce a normal form for languages of the form C^*, with C finite.

▶ **Lemma 17.** *For every $p, q > 0$, finite $C \subseteq \mathbb{2}^*$, and $u_-, u_+ \in C$ so that $\rho(u_-) = \min \rho(C)$ and $\rho(u_+) = \max \rho(C)$, there exists a finite $C' \subseteq C^*$ so that $C^* =_{\text{Rel}} (u_-^p \cup u_+^q)^* \cdot C'$.*

In particular, the lemma implies that $C^* =_{\text{Rel}} (u_- \cup u_+)^* \cdot C'$ for some finite $C' \subseteq C^*$ and u_-, u_+ words of C of minimum and maximum ratio. In other words, it just suffices to iterate two words from C and then append tails of bounded length to obtain the class $\text{Rel}(C^*)$. With this in mind, we can easily prove our characterization for simple languages.

Proof idea of Proposition 7. The left-to-right direction follows from P6 and Lemma 15. For the opposite direction, the case where C is homogeneous is straightforward by P8. For C heterogeneous, we use P5 to we assume wlog that $C = C_1^* \cdots C_n^* u$ and $D = D_1^* \cdots D_m^* v$. Since every C_i is finite (recall that simple languages have star-height 1), we can consider words $w_{i,-}, w_{i,+}$ of minimum and maximum Parikh ratio. Using the normal form of Lemma 17 plus the existence of s.m., we obtain $C_i^* \subseteq_{\text{Rel}} D_{f(i)}^* C_i'$ for a finite $C_i' \subseteq C_i^*$. Thus, $C_1^* \cdots C_n^* u \subseteq_{\text{Rel}} D_{j_1}^* C_1' \cdots D_{j_n}^* C_n' u =_{\text{Rel}} D_{j_1}^* \cdots D_{j_n}^* C_1' \cdots C_n' u \subseteq_{\text{Rel}} D_1^* \cdots D_m^* v$. ◀

7 Regular languages

We now prove the characterization theorem for unions of simple languages. Thanks to this theorem and to Proposition 3, we will obtain an effective characterization for arbitrary regular languages, and thus solve the Class Containment Problem in its full generality.

▶ **Theorem 10.** For finite unions $C = \bigcup_i C_i$ and $D = \bigcup_j D_j$ of simple languages, we have $C \subseteq_{\text{Rel}} D$ if and only if for all i $\pi(C_i) \subseteq \pi(D)$, there is j with $C_i \xrightarrow{s.m.} D_j$ and if C_i is heterogeneous, then $C_i \setminus [D_j]_\pi$ is regular and $C_i \setminus [D_j]_\pi \subseteq_{\text{Rel}} \bigcup_{j' \neq j} D_{j'}$.

Note in particular that the conditions in the characterization of Theorem 10 require that $C_i \setminus [D_j]_\pi$ is regular. Despite that, this property is always verified when $C_i \xrightarrow{s.m.} D_j$ and C_i is heterogeneous by Proposition 12 from Section 5. Indeed, $C_i \xrightarrow{s.m.} D_j$ for C_i heterogeneous implies that all components of C_i are mapped to components of D_j. In view of Lemma 5 and the fact that C_i and D_j have star-height 1, this implies that $\rho(\mathit{cycles}(C_i)) \subseteq \rho(\mathit{cycles}(D_j))$, and hence, by Proposition 12, $C_i \setminus [D_j]_\pi$ is regular. We are now ready to prove the theorem.

Proof of Theorem 10. For the left-to-right implication, by Lemma 8, we have that $C_i \subseteq_{\text{Rel}} D$ for every i. Containment of Parikh images follows then from P6. For any fixed i, if C_i is homogeneous we have $C_i \xrightarrow{s.m.} D_j$ for every j, and if it is smooth heterogeneous, then Lemma 15 yields the existence of some j so that $C_i \xrightarrow{s.m.} D_j$. By Proposition 12, $C_i \setminus [D_j]_\pi$ is regular, and we now prove that $C_i \setminus [D_j]_\pi \subseteq_{\text{Rel}} \bigcup_{j' \neq j} D_{j'}$. Take $R \in \text{Rel}(C_i \setminus [D_j]_\pi)$ and a regular $L \subseteq (C_i \setminus [D_j]_\pi) \otimes \mathbb{A}^*$ so that $[\![L]\!] = R$. Since $C_i \setminus [D_j]_\pi \subseteq C_i$, we have $R \in \text{Rel}(C_i) \subseteq \text{Rel}(D)$, by P1 and hypothesis. Let $L' \subseteq D \otimes \mathbb{A}^*$ be a regular language so that $[\![L']\!] = [\![L]\!] = R$. Since the projection onto $\mathbb{2}$ of L and L' have necessarily the same Parikh image, it follows that $L' \cap (D_j \otimes \mathbb{A}^*) = \emptyset$, and thus that $L' \subseteq (\bigcup_{j' \neq j} D_{j'}) \otimes \mathbb{A}^*$ or, in other words, that $R \in \text{Rel}(\bigcup_{j' \neq j} D_{j'})$.

For the right-to-left implication, for C_i homogeneous, $\pi(C_i) \subseteq \pi(D)$ implies $C_i \subseteq_{\text{REL}} D$ by P8. For C_i heterogeneous, we have $C_i = (C_i \setminus [D_j]_\pi) \cup (C_i \cap [D_j]_\pi)$. By hypothesis plus property P1, $C_i \setminus [D_j]_\pi \subseteq_{\text{REL}} D$. Then, by Lemma 8, it only remains to check that $C_i \cap [D_j]_\pi \subseteq_{\text{REL}} D$. Now, by Proposition 12 and Proposition 3, $C_i \cap [D_j]_\pi$ is $=_{\text{REL}}$-equivalent to a finite union of simple languages $(C'_k)_{k\in K}$. Note that $C'_k \subseteq_{\text{REL}} C_i$ for all $k \in K$. Then, by the left-to-right direction of Proposition 7, we have $C'_k \xrightarrow{s.m.} C_i$ for all k. By composition of synchronizing morphisms, we obtain $C'_k \xrightarrow{s.m.} D_j$ for all $k \in K$. Since we also have that $\pi(C'_k) \subseteq \pi(D_j)$, by the right-to-left direction of Proposition 7, we have that $C'_k \subseteq_{\text{REL}} D_j$ for all $k \in K$. Then, from Lemma 8 it follows that $C_i \subseteq_{\text{REL}} D_j \subseteq D$. Since this happens for every C_i, again by Lemma 8 the statement follows. ◀

8 Decidability and complexity

We have given a characterization of the pairs C, D of regular languages that satisfy $C \subseteq_{\text{REL}} D$. We argue that this characterization is effective.

As explained in Section 7, there are three main steps that one need to take for deciding whether $C \subseteq_{\text{REL}} D$, for two given regular languages C, D: First, one needs to decompose C and D as finite unions $\bigcup_i C_i$ and $\bigcup_j D_j$ of simple languages. This preprocessing relies on two constructions: the computation of the normal form for semi-linear sets and the construction of an automaton for $C \cap [D]_\pi$, proving that is regular. A close inspection of these proofs in Section 5 shows that both procedures are effective, and thus so is the decomposition.

Then, based on the characterization of Theorem 10, one has to identify suitable synchronizing morphisms from each C_i to some D_j. This step boils down to checking whether two components $C^*_{i,i'}$ and $D^*_{j,j'}$ of concat-star languages satisfy $\rho(C^*_{i,i'}) \subseteq \rho(D^*_{j,j'})$. Thanks to the insight of Lemma 5, the containment of Parikh ratios and thus the existence of such synchronizing morphism is decidable.

Finally, the third step uses Theorem 10, reducing the problem $\bigcup_i C_i \subseteq_{\text{REL}} \bigcup_j D_j$ to sub-problems of the form $C_i \setminus [D_{j_i}]_\pi \subseteq_{\text{REL}} \bigcup_{j' \neq j_i} D_{j'}$, which has a smaller union in the right hand-side and thus can be solved recursively (but in principle non-elementary).

The above arguments show that the Class Containment Problem is decidable. Once we know that $C \subseteq_{\text{REL}} D$ for two given regular languages C, D, it is reasonable to ask whether it is possible to resynchronize any relation from C to D, namely, whether there is an algorithm that transforms any automaton $\mathcal{A}$ recognizing $L \subseteq C \otimes \mathbb{A}^*$ into an automaton $\mathcal{A}'$ recognizing $L' \subseteq D \otimes \mathbb{A}^*$ so that $[\![L']\!] = [\![L]\!]$. A close inspection to our decision procedure for $C \subseteq_{\text{REL}} D$ gives a positive answer to the question. Indeed, all our proofs are constructive.

We can summarize the above arguments with the following corollary.

▶ **Corollary 18.** *There is a non-elementary algorithm that, given two regular languages $C, D \subseteq \mathbb{2}^*$, decides whether $C \subseteq_{\text{REL}} D$.*
There is also a non-elementary algorithm that, given an automaton for $L \subseteq C \otimes \mathbb{A}^*$, constructs an automaton for some $L' \subseteq D \otimes \mathbb{A}^*$ so that $[\![L']\!] = [\![L]\!]$, provided $C \subseteq_{\text{REL}} D$.

9 Discussion

The overall picture we obtain from our results is that $\text{REL}(C) \subseteq \text{REL}(D)$ depends on comparing the ratio growth of the two coordinates on the cycles of the transition graph of the automata $\mathcal{A}_C$, $\mathcal{A}_D$ recognizing C, D. Concretely, our reduction into synchronizing morphisms for simple languages can be thought of restricting our attention to cycles $c_1, \ldots, c_n$ of $\mathcal{A}_C$ so that: c_{i+1} is reachable from c_i, and c_i or c_{i+1} is heterogeneous (recall that in a

simple concat-star language, there are no consecutive homogeneous components). Intuitively, $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$ whenever $\pi(C) \subseteq \pi(D)$ and for every sequence of cycles $c_1, \ldots, c_n$ as before, there exists a corresponding sequence of cycles $c'_1, \ldots, c'_n$ in $\mathcal{A}_D$ with the same properties so that c_i and c'_i have the same Parikh ratio for every i.

We also recall (cf. proof of Lemma 15) that our characterization holds for the containment problem $\textsc{Rel}(C) \subseteq \textsc{Rel}(D)$, but also for any variant with a fixed alphabet of cardinality at least 2. For the variant with a unary alphabet $\mathbb{A}$, it is easy to see that $\textsc{Rel}_{\mathbb{A}}(C) \subseteq \textsc{Rel}_{\mathbb{A}}(D)$ is equivalent to $\pi(C) \subseteq \pi(D)$. As concerns relations of higher arity defined by control languages $C \subseteq \mathbb{k}^* = [1, k]^*$, it is not clear if a similar characterization may hold. For example, the normal form of Lemma 11 does not generalize to control alphabets of more than two letters. Finally, we leave for future work the issue of determining the precise complexity of the Class Containment Problem.

References

1. Bahareh Badban and Mohammad Torabi Dashti. Semi-linear parikh images of regular expressions via reduction. In *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 6281 of *Lecture Notes in Computer Science*, pages 653–664. Springer, 2010. doi:10.1007/978-3-642-15155-2_57.
2. Jean Berstel. *Transductions and Context-Free Languages*. B. G. Teubner, 1979.
3. Mikołaj Bojańczyk. Transducers with origin information. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 8573 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2014. doi:10.1007/978-3-662-43951-7.
4. Julius Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.
5. Olivier Carton, Christian Choffrut, and Serge Grigorieff. Decision problems among the main subfamilies of rational relations. *Informatique Théorique et Applications (ITA)*, 40(2):255–275, 2006. doi:10.1051/ita:2006005.
6. Christian Choffrut. Relations over words and logic: A chronology. *Bulletin of the EATCS*, 89:159–163, 2006.
7. Marek Chrobak. Finite automata and unary languages. *Theoretical Computer Science*, 47(3):149–158, 1986. doi:10.1016/0304-3975(86)90142-8.
8. Calvin C. Elgot and Jorge E. Mezei. On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68, 1965. doi:10.1147/rd.91.0047.
9. Diego Figueira and Leonid Libkin. Path logics for querying graphs: Combining expressiveness and efficiency. In *Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 329–340. IEEE Computer Society Press, 2015. doi:10.1109/LICS.2015.39.
10. Diego Figueira and Leonid Libkin. Synchronizing relations on words. *Theory of Computing Systems*, 57(2):287–318, 2015. doi:10.1007/s00224-014-9584-2.
11. Emmanuel Filiot, Ismaël Jecker, Christof Löding, and Sarah Winter. On equivalence and uniformisation problems for finite transducers. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 125:1–125:14. Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.125.
12. Eryk Kopczyński and Anthony Widjaja To. Parikh images of grammars: Complexity and applications. In *Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 80–89. IEEE Computer Society Press, 2010. doi:10.1109/LICS.2010.21.
13. Maurice Nivat. Transduction des langages de Chomsky. *Annales de l'Institut Fourier*, 18:339–455, 1968.