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

1 Introduction

We study 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 [3, 6, 11], with much subsequent work done later (*e.g.*, [1, 5]). 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* [1] or *Synchronous* [5]), 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* [8]. 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 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 [11]). 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 $\mathbb{A} = \{a, b\}$.

* This work is going to appear in the proceedings of ICALP 2018.

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 [4]. 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 [8] that this problem is decidable for some particular instances of D, namely, for $D = \textit{Recognizable}$, *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 [8]. 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 [8]. The formalism of synchronizations has been extended beyond rational relations by means of semi-linear constraints [7] in the context of path querying languages for graph databases.

The paper [2] 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 [9] 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 [2, 9] are of rather different nature than our Class Containment Problem.

2 Characterization of the Class Containment Problem

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

The main idea is to decompose languages as finite unions of what we call *simple* languages. The definition can be found in the article ([10]) but the interest of these languages lies in the fact that we can prove the following two results.

▶ **Proposition 1.** *Every regular language $C \subseteq \mathbb{2}^*$ is effectively $=_{\text{REL}}$-equivalent to a finite union of simple languages.*

The proof of the above proposition is quite involved and it can be found in full detail in the appendix of the article.

▶ **Proposition 2.** *The class containment problem for simple languages C and D is decidable.*

Moreover, Proposition 2 follows from a characterization of class containment for simple languages in terms of their Parikh-images and cycles which is relatively simple.

The generalization of the previous result to the case where C is a finite union of simple languages is given by the following basic result.

▶ **Lemma 3.** $C_1 \cup C_2 \subseteq_{\text{REL}} D$ *iff* $C_1 \subseteq_{\text{REL}} D$ *and* $C_2 \subseteq_{\text{REL}} D$.

The characterization turns out to be more involved when we have unions on the right hand-side. In particular, The analogous of Lemma 3 for unions on the right hand-side does not hold in general.

The characterization we provide is inductive on the number of languages that are unioned on the right hand-side.

References

1 Jean Berstel. *Transductions and Context-Free Languages.* B. G. Teubner, 1979.

2 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.

3 Julius Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

4 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.

5 Christian Choffrut. Relations over words and logic: A chronology. *Bulletin of the EATCS*, 89:159–163, 2006.

6 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.

7 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.

8 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.

9 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.

10 D. Figueira M.E. Descotte and G. Puppis. Resynchronizing classes of word relations. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018. URL: `hal-01721046v2`.

11 Maurice Nivat. Transduction des langages de Chomsky. *Annales de l'Institut Fourier*, 18:339–455, 1968.