Separating Automatic Relations

Pablo Barceló
Institute for Mathematical and Computational Engineering, Universidad Católica de Chile

Diego Figueira
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR5800, F-33400 Talence, France

Rémi Morvan
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR5800, F-33400 Talence, France

Abstract

We study the separability problem for automatic relations (i.e., relations on finite words definable by synchronous automata) in terms of recognizable relations (i.e., finite unions of products of regular languages). This problem takes as input two automatic relations R and R', and asks if there exists a recognizable relation S that contains R and does not intersect R'. We show this problem to be undecidable when the number of products allowed in the recognizable relation is fixed. In particular, checking if there exists a recognizable relation S with at most k products of regular languages that separates R from R' is undecidable, for each fixed $k \geqslant 2$.

2012 ACM Subject Classification Theory of computation → Regular languages

Keywords and phrases Automatic relations, recognizable relations, separability, finite colorability

☞ This pdf contains internal links: clicking on a notion leads to its *definition*.[1]

1 Introduction

Context. The study of classes of relations on words has become an important topic in language theory [11, 22, 5, 12, 8], and also in areas such as databases and verification where they are used to build expressive languages. For instance, classes of relations of this kind are relevant for querying strings over relational databases [3], comparing paths in graph databases [2], or defining string constraints for model checking [19]. The most studied such classes include *recognizable*, *automatic*, and *rational* relations, each one of the latter two strictly extending the previous one. *Rational relations* are those definable by multi-head automata, with heads possibly moving asynchronously; automatic relations are rational relations that are accepted by multi-head automata whose heads are forced to move synchronously; and recognizable relations correspond to finite unions of products of regular languages (or, equivalently, to languages recognized via finite monoids, by Mezei's Theorem). By definition, all of these classes coincide with the class of regular languages when restricted to unary relations.

Prior work has focused on the *Rec-definability problem*, which takes as input an n-ary rational relation R and asks whether it is equivalent to a recognizable relation $\bigcup_i L_{i,1} \times \cdots \times L_{i,n}$, where each $L_{i,j}$ is a regular language. Intuitively, the problem asks whether the different components of the rational relation R are almost independent of one another. The study of Rec-definability is relevant since relations enjoying this property are often amenable to some analysis including, e.g., abstract interpretations in program verification, variable elimination in constraint logic programming, and query processing over constraint databases (see the introduction of [1] for a thorough discussion on this topic).

In general, Rec-definability of rational relations is undecidable, but it becomes decidable for two important subclasses: deterministic rational relations and automatic

[1] This result was achieved by using the `knowledge` package and its companion tool `knowledge-clustering`.

relations. For deterministic rational relations, Rec-definability has been shown to be decidable in double-exponential time for binary relations by Valiant [26]—improving Stearns's triple-exponential bound [24]. The decidability result was later extended to relations of arbitrary arity by Carton, Choffrut and Grigorieff [7, Theorem 3.7]. For automatic relations, the decidability of Rec-definability can be obtained by a simple reduction to the problem of checking whether a finite automaton recognizes an infinite language [20] – which is decidable via a standard reachability argument. The precise complexity of the problem, however, was only recently pinned down. By applying techniques based on Ramsey Theorem over infinite graphs, it was shown that Rec-definability of automatic relations is PSpace-complete when relations are specified by non-deterministic automata [1, Theorem 1] [4, Corollary 2.9].

On the other hand, much less is known about the *Rec-separability problem*, which takes two n-ary rational relations $R, R' \subseteq \mathbb{A}^* \times \mathbb{A}^*$ and checks whether there is a recognizable relation $S = \bigcup_i L_{i,1} \times \cdots \times L_{i,n}$ with $R \subseteq S$ and $R' \cap S = \varnothing$. In other words, this problem asks whether we can *overapproximate* R with a recognizable relation S that is constrained not to intersect with R'. Separability problems of this kind abound in theoretical computer science, in particular in formal language theory where they have gained a lot of attention over the last few years —see, e.g., [23, 15, 10, 9].

As for definability, the Rec-separability problem for rational relations is in general undecidable. In this paper we focus on the separability problem for automatic relations, that is, the restriction of the Rec-separability problem defined above to the case when both R and R' are automatic relations. Notice that when R' is the complement of R this problem boils down to Rec-definability. However, Rec-separability for automatic relations is more general than Rec-definability, and to this day it is unknown whether it is decidable.

Main contributions and technical approach. While we do not solve the separability problem for automatic relations, we report on some significant progress in our understanding of the problem. We start by establishing a tight connection between Rec-separability and the colorability problem for "automatic graphs", which may shed some light on the difficulty of the former problem. An automatic graph [6, 13, 16, 17] is an infinite graph defined on the set of all finite words, whose edge set is described by a binary automatic relation. The regular colorability problem is then checking if the automatic graph defined by a binary automatic relation is finitely colorable, with the restriction that each color is a regular language. Concretely, we show that the Rec-separability problem for binary automatic relations is equivalent, under polynomial time reductions, to the regular colorability problem. Moreover, we introduce a hierarchy $(k\text{-Rec})_{k>0}$ of recognizable relations so that the coloring problem, when restricted to $k > 0$ colors—called k-regular colorability problem— reduces to the separability problem by relations of k-Rec. Concretely:

▶ **Theorem 3.1.** *There are polynomial-time reductions:*

1. *from the Rec-separability problem to the regular colorability problem;*
2. *from the regular colorability problem to the Rec-separability problem; and*
3. *from the k-regular colorability problem to the k-Rec-separability problem, for every $k > 0$.*

Further, the last two reductions are so that the second relation in the instance of the separability problem is the identity Id.

The regular colorability problem seems challenging, and in particular we lack tools for establishing that an automatic graph is finitely colorable; let alone checking that said colors define regular sets. On the other hand, it is easy to see that the k-regular

COLORABILITY PROBLEM is undecidable for each fixed $k > 1$ if we lift the restriction that colors define regular sets, i.e., checking if an automatic graph admits a k-coloring—this has been proved in an unpublished thesis by Köcher [14, Proposition 6.5]. We establish that this undecidability holds even with the restriction on colors being regular sets:

▶ **Theorem 4.4.** *The k-REGULAR COLORABILITY PROBLEM on automatic graphs is undecidable, for every $k \geqslant 2$. This holds also for connected automatic graphs.*

By reduction from the k-REGULAR COLORABILITY PROBLEM we obtain an important consequence for our separability problem: It is undecidable to check if two automatic relations can be separated by a recognizable relation defined by a *fixed* number of unions of products of regular languages. More specifically, fix $k > 0$ and define k-PROD as the class of recognizable relations of the form $S = \bigcup_{1 \leqslant i \leqslant k} L_{i,1} \times \cdots \times L_{i,n}$—this hierarchy is intertwined with the $(k\text{-REC})_{k>0}$ hierarchy introduced previously. We show that the k-PROD-SEPARABILITY PROBLEM, *i.e.*, the problem of checking separability for binary automatic relations R and R' in terms of a recognizable relation S in the class k-PROD, is undecidable for $k \geqslant 2$.

▶ **Theorem 5.6.** *The k-PROD-SEPARABILITY PROBLEM is undecidable, for every $k \geqslant 2$.*

At this point, a natural question is whether our decision of restricting the study to the class k-PROD, for fixed $k > 1$, is not too strong, in the sense that it turns undecidable not only the separability but also the definability problem for automatic relations. We show that this is not the case; in fact, by using a simple adaptation of the proof techniques in [1] we can show that the problem of checking if an automatic relation can be expressed as a recognizable relation in k-PROD, for a fixed $k > 0$, is decidable in single-exponential time:

▶ **Corollary 6.4.** *The k-PROD-DEFINABILITY PROBLEM is decidable, for every $k > 0$.*

Remark. For simplicity, we focus on binary automatic relations only. Extending the decidability results to n-ary automatic relations, for $n > 2$ is direct by applying tools in [1].

2 Preliminaries

Automatic and recognizable relations. Let $\mathbb{A}$ be a finite alphabet. We write $\mathbb{A}_\perp$ for the extension of $\mathbb{A}$ with a fresh symbol $\perp$. Given a pair $(w_1, w_2) \in \mathbb{A}^* \times \mathbb{A}^*$, we write $w_1 \otimes w_2$ for the word over alphabet $\mathbb{A}_\perp \times \mathbb{A}_\perp$ that is obtained as follows: first, padding the shorter word with $\perp$'s until both words are of the same length, and then reading the two words synchronously as if they were a single word over a binary alphabet. For example, if $w_1 = aaba$ and $w_2 = aa$, then $w_1 \otimes w_2 = (a,a)(a,a)(b,\perp)(a,\perp)$. For any relation $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$, let us write $\otimes R$ to denote the set

$$\otimes R \ \hat{=}\ \{u \otimes v \mid (u,v) \in R\} \ \subseteq\ (\mathbb{A}_\perp \times \mathbb{A}_\perp)^*.$$

We then have the following:

- $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$ is an *automatic relation* iff $\otimes R$ is a regular language;
- $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$ is a *recognizable relation* iff $R = \bigcup_{i=1}^{n} A_i \times B_i$, where $n \in \mathbb{N}$ and all the A_i's and B_i's are regular languages over $\mathbb{A}$.

We denote by REC the class of all recognizable relations.

▶ **Example 2.1.** For any fixed constant $c > 0$, the relation R composed by all pairs of words of the form (a^n, a^{n+c}), for $n \geqslant 0$, is automatic. In turn, R is not recognizable. An example of a non-automatic relation is the one consisting of all pairs of the form $(a^n, a^{d \cdot n})$, for $n > 0$, for any constant $d > 1$. ◀

Separability. Let R and R' be automatic relations over an alphabet $\mathbb{A}$. A recognizable relation S over $\mathbb{A}$ *separates* R from R' if $R \subseteq S$ and $R' \cap S = \varnothing$.

▶ **Example 2.2.** Consider the automatic relations $R = \{(a^n, a^{n+1}) \mid n \geqslant 0\}$ and $R' = \{(a^n, a^{n+2}) \mid n \geqslant 0\}$. They are separable by the recognizable relation

$$S = (A_{\text{even}} \times A_{\text{odd}}) \cup (A_{\text{odd}} \times A_{\text{even}}),$$

where A_{even} and A_{odd} are the regular languages $(aa)^*$ and $a(aa)^*$, respectively. ◀

We study the following separability problem, for a class $\mathcal{C}$ of recognizable relations.

Problem: $\mathcal{C}$-*SEPARABILITY PROBLEM*
Input: Automatic relations R and R' over $\mathbb{A}$
Question: Is there a recognizable relation in $\mathcal{C}$ over $\mathbb{A}$ that separates R from R'?

We also consider the $\mathcal{C}$-*DEFINABILITY PROBLEM*, which takes as input an automatic relation R and asks if there is a recognizable relation S with $S = R$. It is easy to see that the $\mathcal{C}$-DEFINABILITY PROBLEM corresponds to an instance of the $\mathcal{C}$-SEPARABILITY PROBLEM.

▷ Fact 2.3. For any class $\mathcal{C}$ of recognizable relations, the $\mathcal{C}$-DEFINABILITY PROBLEM is Turing-reducible to the $\mathcal{C}$-SEPARABILITY PROBLEM.

Proof. Reduce an instance R of the DEFINABILITY PROBLEM to the instance $(R, (\mathbb{A}^* \times \mathbb{A}^*) \setminus R)$ of the SEPARABILITY PROBLEM. ◀

The following is known regarding the complexity of the REC-definability problem.

▶ **Proposition 2.4.** [1, Theorem 1] *The* REC-DEFINABILITY PROBLEM *for automatic relations specified by non-deterministic automata is* PSPACE*-complete.*

Automatic graphs. Let R be a binary automatic relation over $\mathbb{A}$. This relation naturally defines a directed graph $\mathcal{G}_R = (\mathbb{A}^*, R)$, *i.e.*, the nodes of $\mathcal{G}_R$ are the words over $\mathbb{A}^*$ and there is an edge in $\mathcal{G}_R$ from word u to word v iff $(u, v) \in R$. An *automatic graph* is a graph of the form $\mathcal{G}_R$, for R an automatic relation. A *k-coloring* of $\mathcal{G}_R$ is a partition of $\mathbb{A}^*$ into k sets $V_1, \ldots V_k$ such that $(V_i \times V_i) \cap E = \varnothing$ for every i.

▶ **Example 2.5.** Consider again the automatic relation $R = \{(a^n, a^{n+c}) \mid n \geqslant 0\}$, where $c > 0$ is a fixed constant. The graph $\mathcal{G}_R$ is formed by a disjoint union of c infinite directed paths, and thus it is 2-colorable. ◀

A *k-regular coloring* of $\mathcal{G}_R$ is a k-coloring whose colors $(V_i)_{1 \leqslant i \leqslant k}$ are regular languages. A *regular coloring* is a k-regular coloring for some k.

▶ **Example 2.6.** The automatic graph $\mathcal{G}_R$ from Example 2.5 is 2-regular colorable. In fact, it suffices to define color V_1 as having every word of the form a^n with $n \equiv i \pmod{2c}$, for $i \in [0, c-1]$, and $V_2 = \mathbb{A}^* \setminus V_1$. ◀

The *k-REGULAR COLORABILITY PROBLEM* is the problem of whether $\mathcal{G}_R$ has a k-regular coloring, given an automatic relation R. The *REGULAR COLORABILITY PROBLEM* is the problem of whether $\mathcal{G}_R$ has a regular coloring.

3 Separability is Equivalent to Regular Colorability

We start by showing that the SEPARABILITY PROBLEM in terms of arbitrary recognizable relations is equivalent, under polynomial time reductions, to the REGULAR COLORABILITY PROBLEM. To make our statement precise, we need some terminology introduced below. Let *k-*REC be the class of languages expressed by unions of products of k regular languages which form a partition, that is (in the binary case), relations of the form $(L_{i_1} \times L_{j_1}) \cup \cdots \cup (L_{i_\ell} \times L_{j_\ell})$ for some regular partition $L_1, \dots, L_k$ of $\mathbb{A}^*$ and $\ell \in \mathbb{N}$. Note that $\text{REC} = \bigcup_k k\text{-REC}$. Let us denote by *Id* the identity relation (on any implicit alphabet). Observe that *Id* is automatic but not recognizable.

▶ **Theorem 3.1.** *There are polynomial-time reductions:*

1. *from the* REC-SEPARABILITY PROBLEM *to the* REGULAR COLORABILITY PROBLEM*;*
2. *from the* REGULAR COLORABILITY PROBLEM *to the* REC-SEPARABILITY PROBLEM*; and*
3. *from the k-*REGULAR COLORABILITY PROBLEM *to the k-*REC-SEPARABILITY PROBLEM*, for every $k > 0$.*

Further, the last two reductions are so that the second relation in the instance of the SEPARABILITY PROBLEM *is the identity Id.*

Proof. We start with the last two reductions. Given an automatic graph $\mathcal{G}_E$ over an alphabet $\mathbb{A}$, consider the instance R_1, R_2 for the REC-SEPARABILITY PROBLEM, where $R_1 = E$ and $R_2 = Id$. If $\mathcal{G}_E$ is k-regular colorable via the coloring $V_1, \dots, V_k$ then the k-REC relation $\bigcup_{i \neq j} V_i \times V_j$ separates R_1 and R_2. Conversely, if a k-REC relation $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$ on the partition $V_1 \dot\cup \cdots \dot\cup V_k = \mathbb{A}^*$ separates R_1 and R_2, then $\bigcup_{i \neq j} V_i \times V_j$ also separates R_1 and R_2, and this implies that $V_1, \dots, V_k$ is a k-coloring for $\mathcal{G}_E$.

For the first reduction, let us introduce some terminology. Given two relations R_1, R_2 over $\mathbb{A}^*$, say that $u \in \mathbb{A}^*$ is *compatible* with $u' \in \mathbb{A}^*$ when for all word $v \in \mathbb{A}^*$:

$$\begin{array}{lll} (\text{COMP}_l)\text{: } (u,v) \in R_1 \Rightarrow (u',v) \notin R_2, & & (\text{COMP}_r)\text{: } (v,u) \in R_1 \Rightarrow (v,u') \notin R_2, \\ (\text{COMP}'_l)\text{: } (u',v) \in R_1 \Rightarrow (u,v) \notin R_2 & \text{and} & (\text{COMP}'_r)\text{: } (v,u') \in R_1 \Rightarrow (v,u) \notin R_2. \end{array}$$

Define the *incompatibility graph* $\mathcal{I}nc_{R_1,R_2}$ as the graph whose vertices are all words of $\mathbb{A}^*$, and with an edge from u to v whenever u is not compatible with v. Note that $\mathcal{I}nc_{R,Id}$ is exactly the graph $\mathcal{G}_R$. For a less trivial example of an incompatibility graph, see Appendix B.

▶ **Lemma 3.2.** *If R_1 and R_2 are automatic, then so is $\mathcal{I}nc_{R_1,R_2}$. Moreover, we can build an automaton for $\mathcal{I}nc_{R_1,R_2}$ in polynomial time in the size of the automata for R_1 and R_2.*

See the proof of Lemma 3.2 in page 15.

Given an instance (R_1, R_2) of the SEPARABILITY PROBLEM, we reduce it to the REGULAR COLORABILITY PROBLEM on its incompatibility graph $\mathcal{I}nc_{R_1,R_2}$.

Left-to-right implication: Assume that there exists S in k-REC that separates R_1 from R_2. Then S can be written as $(A_{i_1} \times A_{j_1}) \cup \cdots \cup (A_{i_\ell} \times A_{j_\ell})$, where $(A_1, \dots, A_k)$ is a partition of $\mathbb{A}^*$ in k regular languages. We define the color of a word $u \in \mathbb{A}^*$ as the unique $i \in [\![1, k]\!]$ *s.t.* $u \in A_i$. In other words, the coloring is simply $(A_1, \dots, A_k)$.

This is indeed a proper coloring: if u and u' have the same color, we claim that u is compatible with u'. In fact, take any $v \in \mathbb{A}^*$. If $(u,v) \in R_1$, then $(u,v) \in S$, so $(u,v) \in A_{i_m} \times A_{j_m}$ for some m. But since u has the same color as u', the fact that $u \in A_{i_m}$ implies $u' \in A_{i_m}$, and hence $(u',v) \in A_{i_m} \times A_{j_m} \subseteq S$. But S separates R_1 from R_2, and therefore $(u',v) \notin R_2$. This tells us that (COMP_l) holds. The other conditions hold by symmetry. We conclude that $(A_1, \dots, A_k)$ defines a proper coloring of $\mathcal{I}nc_{R_1,R_2}$, and this coloring, with k colors, is regular since the A_i's are regular languages by definition.

Right-to-left implication: Assume that $\mathcal{I}nc_{R_1,R_2}$ is finitely colorable, say by $(A_1, \dots, A_k)$. Then let S be the union of all S_i's where

$$\begin{aligned} S_i \mathrel{\hat=} \{&(u,v) \mid u \in A_i \text{ and } (u',v) \in R_1 \text{ for some } u' \in A_i\} \\ &\cup \{(u,v) \mid v \in A_i \text{ and } (u,v') \in R_1 \text{ for some } v' \in A_i\}. \end{aligned}$$

Since $(A_1, \dots, A_k)$ covers every node of $\mathcal{I}nc_{R_1,R_2}$, we get $R_1 \subseteq S$. Moreover, we claim that $R_2 \cap S = \varnothing$. Indeed, if $(u,v) \in S$, then $(u,v) \in S_i$ for some i, j. It either means that 1 $(u',v) \in R_1$ for some $u' \in A_i$, or 2 $(u,v') \in R_2$ for some $v' \in A_i$. In case 1 , the fact that $u \in A_i$ implies that u and u' have the same color. Thus, u must be compatible with u' and hence $(u,v) \notin R_2$ using (comp$'_l$). The other case is symmetric. Therefore, $(u,v) \notin R_2$, and thus S separates R_1 from R_2.

Finally, S is recognizable; in fact, $S = \bigcup_{i=1}^{k} \big(A_i \times R_1[A_i]\big) \cup \big(R_1^{-1}[A_i] \times A_i\big)$, where for any set $X \subseteq \mathbb{A}^*$ we define $R_1[X]$ (resp. $R_1^{-1}[X]$) as the set of $v \in \mathbb{A}^*$ (resp. $u \in \mathbb{A}^*$) such that $(u,v) \in R_1$ for some $u \in X$ (resp. $v \in X$). Hence, R_1 and R_2 are REC-separable. ◀

It is not known to date whether the REGULAR COLORABILITY PROBLEM is decidable, and hence the same holds for the REC-SEPARABILITY PROBLEM in light of the previous theorem. This is due to the fact that there are no known characterizations of when an automatic graph is finitely colorable. In spite of this, we believe that the connection between separability and finite colorability is of interest, as it provides us with a way to define and study meaningful restrictions of our problems. The first such a restriction corresponds to the k-REGULAR COLORABILITY PROBLEM for automatic graphs, which we study in the next section.

4 k-Regular Colorability Problem

While we do not know how to approach the REGULAR COLORABILITY PROBLEM, we show that as soon as we add the restriction that the number of colors be bounded, the problem becomes undecidable; i.e., the k-REGULAR COLORABILITY PROBLEM is undecidable for $k \geqslant 2$. Using this, we obtain in the next section the undecidability for the SEPARABILITY PROBLEM on two natural classes of recognizable relations. This is proven by a reduction from a suitable problem on reversible Turing Machines with certain restrictions, which we call "well-founded".

4.1 Regularity of Reachability for Turing Machines

We use the standard notation $u[i..j]$ to denote the factor of a word u between (and including) positions i and j, and $u[i]$ to denote $u[i..i]$. Consider any deterministic Turing Machine (TM) $T = \langle Q, \Gamma, \bot, \delta, q_0, F\rangle$, where Q is the set of states, Γ is tape alphabet, $\bot$ is the blank symbol, $\delta : (Q \setminus F) \times \Gamma_\bot \to Q \times \Gamma \times \{L, R\}$ is the transition (partial) function, where $\Gamma_\bot = \Gamma \cup \{\bot\}$, and q_0 and F is the initial and set of final states, respectively. We represent a configuration with tape content $w \cdot \bot^\omega$ (where $w \in \Gamma^* \cdot \{\bot\}$), in state q and with the head pointing to the cell number $1 \leqslant i \leqslant |w|$, as the string

$$w[1..i-1] \cdot (w[i], q) \cdot w[i+1..|w|]$$

over the alphabet $\mathbb{A}_T = \Gamma \cup (\Gamma_\bot \times Q)$. In light of this representation, we will henceforth denote by "configuration" any string from the set $\mathit{Confs}_T \mathrel{\hat=} (\Gamma^* \cdot (\Gamma_\bot \times Q)) \cup (\Gamma^* \cdot (\Gamma \times Q) \cdot \Gamma^*)$. The *initial configuration* is $(\bot, q_0)$. The *configuration graph* of T is the infinite graph $\mathcal{G}^T$ having Confs_T as set of vertices and an edge from c to c', denoted $c \to c'$, if c' is the configuration of the next step of T starting from c. Observe that the configuration graph $\mathcal{G}^T$ of any TM T is an effective automatic graph (see, e.g., [17]).

We say that a deterministic TM T is *reversible* if every node of $\mathcal{G}^T$ has in-degree at most 1, in other words if the machine is co-deterministic. We say that a TM T is a *well-founded Reversible Turing Machine* (*wf-RTM*) if its configuration graph is such that (1) the initial configuration has in-degree 0 (2) every node has in-degree and out-degree at most one (3) there are no infinite backward paths $c_1 \leftarrow c_2 \leftarrow \cdots$ in $\mathcal{G}^T$.

Note that every well-founded Reversible Turing Machine is deterministic and reversible and, moreover, its configuration graph is a (possibly infinite) disjoint union of directed paths, which are all finite, or isomorphic to $(\mathbb{N}, +1)$. The set of *reachable configurations*, denoted by Reach, is the set of all configurations that admit a path from the initial configuration in $\mathcal{G}^T$, for a given TM T. Such a configuration graph is depicted on Figure 1a.

The *REACHABLE REGULARITY PROBLEM* is the problem of, given a wf-RTM T, whether its set of reachable configurations is a regular language. To show that is it undecidable, we exhibit a reduction from the halting problem on deterministic reversible Turing machines.

▶ **Proposition 4.1** ([18, Theorem 1]). *The halting problem on deterministic reversible Turing machines is undecidable.*

For more details and pointers on reversible Turing machines, see [21, Chapter 5].

▶ **Lemma 4.2.** *The* REACHABLE REGULARITY PROBLEM *is undecidable.*

See the proof of Lemma 4.2 in page 15.

Proof sketch. By reducing the halting problem on deterministic reversible Turing machines, in such a way that the reachable configurations whose state q coincide with the state of the original machine are of the form $(uqva^nb^n)$ where (uqv) is a configuration of the original machine, a and b are new symbols, and $n \in \mathbb{N}$. Transitions are defined in such a way that the new machine is a wf-RTM, and moreover:

- if the original machine was halting, then the reachable configurations of the new one is finite and hence regular;
- otherwise, the set of reachable configurations is not regular, which follows from the non-regularity of any infinite subset of $\{a^nb^n \mid n \in \mathbb{N}\}$.

More details and an illustration can be found in Appendix A. ◀

4.2 Undecidability of the k-Regular Colorability Problem

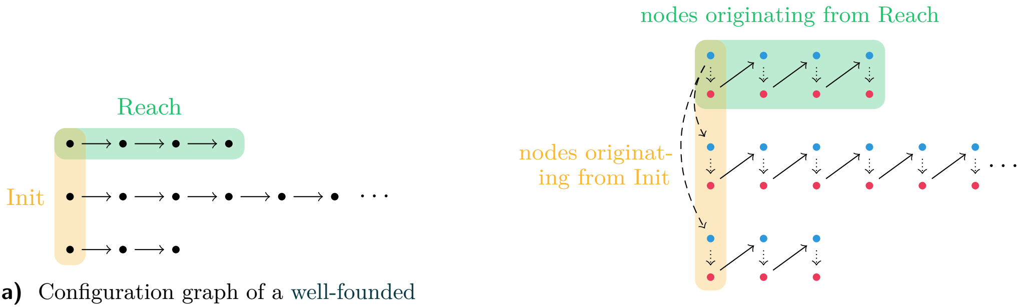
We can now show undecidability for the k-REGULAR COLORABILITY PROBLEM by reduction from the REACHABLE REGULARITY PROBLEM as defined before.

▷ Fact 4.3. Given an automatic graph, the set of nodes with no predecessor is effectively a regular language.

▶ **Theorem 4.4.** *The* k-REGULAR COLORABILITY PROBLEM *on automatic graphs is undecidable, for every* $k \geqslant 2$. *This holds also for connected automatic graphs.*

Proof. By reduction from the REACHABLE REGULARITY PROBLEM for wf-RTMs (Lemma 4.2). We first show it for $k = 2$. Given a wf-RTM T, let c_{init} be its initial configuration. Observe that the set Init of all vertices of $\mathcal{G}^T$ with in-degree 0 is an effective regular language (by Fact 4.3), and that $c_{init} \in$ Init. Let B and R be fresh symbols. Consider the automatic graph $\mathcal{G}_E$ having an edge from (z, c) to (z', c') if either

1. $(z, z') = (B, R)$ and $c = c'$;
2. $(z, z') = (R, B)$ and there is an edge from c to c' in $\mathcal{G}^T$; or
3. $(z, z') = (B, B)$, $c = c_{init}$ and $c' \in \text{Init} \setminus \{c_{init}\}$.



(a) Configuration graph of a well-founded Reversible Turing Machine.

(b) The automatic graph to which it is reduced.

Figure 1 Reduction used in the proof of Theorem 4.4.

Fresh symbols B and R are utilized to represent two versions of each configuration - one in Blue and one in Red. This graph is depicted on Figure 1. Note that $\mathcal{G}_E$ is connected and 2-colorable: in fact, it is a directed (possibly infinite) tree with root (B, c_{init}).

We claim that $\mathcal{G}_E$ is 2-regular colorable if, and only if, the set of reachable configurations of T is a regular language. In fact, up to permuting the two-colors, $\mathcal{G}_E$ admits a unique 2-coloring, defined by:

$$C_1 \quad \hat{=} \quad \{B\} \times \mathrm{Reach} \quad \cup \quad \{R\} \times (\mathit{Confs}_T \setminus \mathrm{Reach})$$

and C_2 is the complement of C_1. If Reach is regular, then so is C_1. Dually, if C_1 is regular, then Reach is the set of configurations c such that $(B, c) \in C_1$ and hence is regular. It follows that $\mathcal{G}_E$ is 2-regular colorable if and only if the reachable configurations of T are regular, which concludes the proof for $k = 2$.

To prove the statement for any $k > 2$, we define $\mathcal{G}_{E_k}$ as the result of adding a $(k-2)$-clique to $\mathcal{G}_E$ and adding an edge from every vertex of the clique to every vertex incident to an edge of E. This forces the clique to use $k-2$ colors that cannot be used in the remaining part of the graph and the proof is then analogous. ◀

Note that this reduction provides an easy way of building graphs in the shape of Figure 1b that are 2-colorable (in fact, they are trees) but not 2-regular colorable. In fact, we can provide a slightly more direct construction.

▶ **Example 4.5.** On the alphabet $\mathbb{A} = \{a, b\}$, the tree $\mathcal{T}$ whose set of vertices is $V = a^*b^*$ and whose set of edges is $E = E_{\mathrm{incr}} \cup E_{\mathrm{init}}$, with

$$E_{\mathrm{incr}} = \{(a^p b^q,\ a^{p+1} b^{q+1}) \mid p, q \in \mathbb{N}\}$$
$$E_{\mathrm{init}} = \{(\varepsilon,\ a^p) \mid p \in \mathbb{N}\} \cup \{(\varepsilon,\ b^q) \mid q \in \mathbb{N}\},$$

is automatic but not 2-regular colorable: indeed, its only 2-coloring consists in partitioning the vertices of $\mathcal{T}$ into

$$C = \{a^n b^n \mid n \in 2\mathbb{N}\} \cup \{a^p b^q \mid p > q \text{ and } q \text{ is odd}\} \cup \{a^p b^q \mid p < q \text{ and } p \text{ is odd}\}$$

and its complement $V \setminus C$. Let $P = \{a^p b^q \mid p, q \in 2\mathbb{N}\} = (aa)^*(bb)^*$: P is regular, yet $C \cap P = \{a^n b^n \mid n \in 2\mathbb{N}\}$ is not. Hence, C is not regular, and thus $\mathcal{T}$ is not 2-regular colorable. ◀

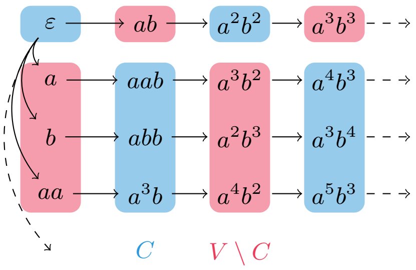


Figure 2 The automatic tree $\mathcal{T}$ of Example 4.5, and its unique 2-coloring $(C, V \setminus C)$, which is not regular.

5 Separability for Bounded Recognizable Relations

In this section we capitalize on the undecidability result of the previous section, showing how this implies the undecidability for the SEPARABILITY PROBLEM on two natural classes of bounded recognizable relations, namely: k-REC, and k-PROD. Remember that, for any k, k-PROD is the subclass of REC consisting of unions of k cross-products of regular languages (which is a subclass of 2^{2k}-REC).

k-Rec-separability. First, observe that the 1-REC-SEPARABILITY PROBLEM is trivially decidable, since the only possible separator is $\mathbb{A}^* \times \mathbb{A}^*$. However, for any other $k > 1$, the problem is undecidable.

▶ **Proposition 5.1.** *The k-REC-SEPARABILITY PROBLEM is undecidable, for every $k > 1$.*

Proof. A consequence of the reduction from the k-REGULAR COLORABILITY PROBLEM of Theorem 3.1, combined with the undecidability of the latter for every $k > 1$ (Theorem 4.4). ◀

k-Prod-separability. On the k-PROD hierarchy we will find the same phenomenon. In particular the case $k = 1$ is also trivially decidable.

▶ **Proposition 5.2.** *The 1-PROD-SEPARABILITY PROBLEM is decidable.*

Proof. Given two automatic relations R_1, R_2, there exists $S \in$ 1-PROD that separates R_1 from R_2 if and only if $\pi_1(R_1) \times \pi_2(R_1)$ separates R_1 from R_2. ◀

As soon as $k > 1$, the k-PROD SEPARABILITY PROBLEM becomes undecidable. This is a consequence of the following simple lemma.

▶ **Lemma 5.3.** *A symmetric automatic relation R and the identity Id are separable by a relation in 2-PROD iff they have a separator of the form $(A \times B) \cup (B \times A)$.*

Proof. Assume that $S \in$ 2-PROD separates R from Id. Then $R \subseteq S$, but since R is symmetric, $R = R^{-1} \subseteq S^{-1}$ so $R \subseteq S \cap S^{-1}$, and hence $R \subseteq S \cap S^{-1}$. Moreover, since S has a trivial intersection with Id, so does $S \cap S^{-1}$. Hence, $S \cap S^{-1}$ separates R from Id.

Since $S \in$ 2-PROD, there exists $A_1, A_2, B_1, B_2 \subseteq \mathbb{A}^*$ such that $S = A_1 \times B_1 \cup B_2 \times A_2$. Note that $S \cap Id = \varnothing$ yields $A_i \cap B_i = \varnothing$ for each $i \in \{1, 2\}$. Finally:

$$\begin{aligned} S \cap S^{-1} &= \big(A_1 \times B_1 \cup B_2 \times A_2\big) \cap \big(B_1 \times A_1 \cup A_2 \times B_2\big) \\ &= \big((A_1 \times B_1) \cap (B_1 \times A_1)\big) \cup \big((A_1 \times B_1) \cap (A_2 \times B_2)\big) \end{aligned}$$

$$\begin{aligned}
&\cup \big((B_2 \times A_2) \cap (B_1 \times A_1)\big) \cup \big((B_2 \times A_2) \cap (A_2 \times B_2)\big)\\
&= \overbrace{\big((A_1 \cap B_1) \times (A_1 \cap B_1)\big)}^{=\varnothing} \cup \big((A_1 \cap A_2) \times (B_1 \cap B_2)\big)\\
&\quad \cup \big((B_1 \cap B_2) \times (A_1 \cap A_2)\big) \cup \underbrace{\big((A_2 \cap B_2) \times (A_2 \cap B_2)\big)}_{=\varnothing}\\
&= \big((A_1 \cap A_2) \times (B_1 \cap B_2)\big) \cup \big((B_1 \cap B_2) \times (A_1 \cap A_2)\big).
\end{aligned}$$ ◀

We can then establish the following:

▶ **Corollary 5.4.** *A symmetric automatic relation R and Id are separable by a relation in 2-*PROD *iff $\mathcal{G}_R$ is 2-regular colorable.*

Proof. By observing that for any symmetric relation $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$, we have that $A, B \subseteq \mathbb{A}^*$ is a coloring of $\mathcal{G}_R$ if, and only if, $(A \times B) \cup (B \times A)$ separates R from Id. ◀

We can now easily show undecidability for the 2-PROD SEPARABILITY PROBLEM by reduction from the 2-REGULAR COLORABILITY PROBLEM.

▶ **Lemma 5.5.** *The 2-*PROD*-SEPARABILITY PROBLEM is undecidable.*

Proof. By reduction from the 2-REGULAR COLORABILITY PROBLEM on automatic graphs, which is undecidable by Theorem 4.4. Let $\mathcal{G}_R$ be an automatic graph and $\mathcal{G}_{R'}$ the symmetric closure of $\mathcal{G}_R$. It follows that $\mathcal{G}_{R'}$ is still automatic and that there is a 2-regular coloring for $\mathcal{G}_{R'}$ iff there is a 2-regular coloring for $\mathcal{G}_R$ (the same coloring in fact). Thus, by Corollary 5.4, $\mathcal{G}_R$ is 2-regular colorable iff there is a 2-PROD relation that separates $\mathcal{G}_{R'}$ from Id. ◀

Further, this implies undecidability for every larger k:

▶ **Theorem 5.6.** *The k-*PROD*-SEPARABILITY PROBLEM is undecidable, for every $k \geqslant 2$.*

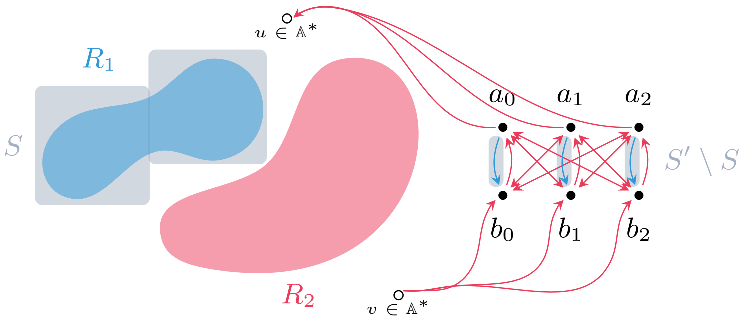


Figure 3 Construction in the proof of Theorem 5.6 for $k = 5$. S is depicted as the union of two (gray) rectangles since $S \in$ 2-PROD. The relation R'_1 is obtained from R_1 (blue shape) by adding all blue edges, namely (a_i, b_i) for $1 \leqslant i \leqslant k-2$. The relation R'_2 is obtained from R_2 (red shape) by adding all red edges, namely every other edge involving a vertex a_i or b_i. Finally, S' (five gray rectangles) is obtained from S by adding each $\{a_i\} \times \{b_i\}$.

Proof. The case $k = 2$ is shown in Lemma 5.5, so suppose $k > 2$. The proof goes by reduction from the 2-PROD-SEPARABILITY PROBLEM. Let R_1, R_2 be a pair of automatic relations over an alphabet $\mathbb{A}$. Consider the alphabet extended with $2(k-2)$ fresh symbols $\mathbb{A}' = \mathbb{A} \mathbin{\dot\cup} \{a_1, \dots, a_{k-2}, b_1, \dots, b_{k-2}\}$. We build automatic relations R'_1, R'_2 over $\mathbb{A}'$ such that (R_1, R_2) are 2-PROD separable over $\mathbb{A}$ iff (R'_1, R'_2) are k-PROD separable over $\mathbb{A}'$.

Let $R'_1 = R_1 \dot\cup \{(a_i, b_i) : 1 \leqslant i \leqslant k-2\}$ and

$$\begin{aligned} R'_2 = R_2 \dot\cup &\{(a_i, w) : w \in \mathbb{A}^*, 1 \leqslant i \leqslant k-2\} \dot\cup \\ &\{(w, b_i) : w \in \mathbb{A}^*, 1 \leqslant i \leqslant k-2\} \dot\cup \\ &\{(a_i, b_j) : 1 \leqslant i, j \leqslant k-2, i \neq j\} \dot\cup \\ &\{(b_i, a_j) : 1 \leqslant i, j \leqslant k-2\} \end{aligned}$$

If (R_1, R_2) has a 2-Prod separator S, then $\tilde{S} \dot\cup \{(a_i, b_i) : 1 \leqslant i \leqslant k-2\}$ is a k-Prod separator of (R'_1, R'_2).

Conversely, if $S' = (A_1 \times B_1) \cup \cdots \cup (A_k \times B_k)$ is a k-Prod separator of (R'_1, R'_2), then for every i there must be some j_i such that $A_{j_i} \times B_{j_i}$ contains (a_i, b_i). Observe that

- $A_{j_i} \cup B_{j_i}$ cannot contain any $a_{i'}$ or $b_{i'}$ for $i' \neq i$, and
- $A_{j_i} \cup B_{j_i}$ cannot contain any $w \in \mathbb{A}^*$;

since otherwise we would have $(A_{j_i} \times B_{j_i}) \cap R'_2 \neq \varnothing$. Hence, $\{i \mapsto j_i\}_i$ is injective, and thus S' is of the form $S' = (A_1 \times B_1) \cup (A_2 \times B_2) \cup (\{a_1\} \times \{b_1\}) \cup \cdots \cup (\{a_{k-2}\} \times \{b_{k-2}\})$. We can further assume that A_1, B_1, A_2, B_2 do not contain any a_i or b_i since otherwise we can remove them preserving the property of being a k-Prod separator of R'_1 and R'_2. Hence, $S \mathrel{\hat=} (A_1 \times B_1) \cup (A_2 \times B_2)$ must cover R_1 and be disjoint from R_2, obtaining that S is a 2-Prod separator of R_1 and R_2. ◀

6 Definability for Bounded Recognizable Relations

Up until now, we have examined two hierarchies of bounded recognizable relations, namely k-Prod and k-Rec. Our previous analysis demonstrated that, for any element in these hierarchies (where $k > 1$), the separability problem is undecidable. Nevertheless, we will now establish that the definability problem is decidable.

Given an automatic relation $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$, consider the automatic equivalence relation $\sim_R \subseteq \mathbb{A}^* \times \mathbb{A}^*$, defined as $w \sim_R w'$ if for every $v \in \mathbb{A}^*$ we have

1. $(w, v) \in R$ iff $(w', v) \in R$, and
2. $(v, w) \in R$ iff $(v, w') \in R$.

It turns out that equivalence classes of $\sim_R$ define the coarsest partition onto which R can be recognized in terms of k-Rec:

▶ **Lemma 6.1.** *For every automatic $R \subseteq \mathbb{A}^* \times \mathbb{A}^*$, $\sim_R$ has index at most k if, and only if, R is in k-Rec.*

Proof. Left-to-right Assume that $\sim_R$ has the equivalence classes $E_1, \ldots, E_k$. Consider the set $P \subseteq \{1, \ldots, k\}^2$ of all pairs (i, j) such that there are $u_i \in E_i$ and $u_j \in E_j$ with $(u_i, u_j) \in R$. Define the k-Rec relation $R' = \bigcup_{(i,j) \in P} E_i \times E_j$. We claim that $R = R'$. In fact, by definition of $\sim_R$, note that if there are $u_i \in E_i$ and $u_j \in E_j$ with $(u_i, u_j) \in R$, then $E_i \times E_j \subseteq R$. Hence, $R' \subseteq R$. On the other hand, for every pair $(u, v) \in R$ there is $(i, j) \in P$ such that $u \in E_i$, $v \in E_j$ implying $(u, v) \in R'$. Hence, $R \subseteq R'$.

Right-to-left If R is a union of products of sets from the partition $E_1 \dot\cup \cdots \dot\cup E_k = \mathbb{A}^*$, then every two elements of each E_i are $\sim_R$-related, and thus $\sim_R$ has index at most k. ◀

We can then conclude that the definability problem for k-Rec is decidable.

▶ **Corollary 6.2.** *The k-Rec-definability problem is decidable, for every $k > 0$.*

Proof. An automatic relation R is in k-Rec iff $\sim_R$ has at most k equivalence classes by Lemma 6.1. In other words, an automatic relation R is not in k-Rec iff the complement of $\sim_R$ contains a $(k+1)$-clique, which can be easily tested. ◀

The relation $\sim_R$ can also be used to characterize which automatic relations are definable in the class k-Prod.

▶ **Lemma 6.3.** *An automatic relation R is in k-Prod if, and only if, $R = (A_1 \times B_1) \cup \cdots \cup (A_k \times B_k)$ where each A_i and B_i is a union of equivalence classes of $\sim_R$.*

Proof. It suffices to show that for every equivalence class E from $\sim_R$, if $A_1 \cap E \neq \varnothing$ then $R = ((A_1 \cup E) \times B_1) \cup \cdots \cup (A_k \times B_k)$, and similarly for B_1. Assume $w \in A_1 \cap E$ and take any pair $(u, v) \in E \times B_1$. We show that $(u, v) \in R$. By definition of $\sim_R$, since $(w, v) \in R$ and $w \sim_R u$, we have that $(u, v) \in R$. ◀

Again, this characterization allows us to show that definability in the class k-Prod is decidable.

▶ **Corollary 6.4.** *The k-Prod-definability problem is decidable, for every $k > 0$.*

Proof. By brute force testing whether the automatic relation R is equivalent to $(A_1 \times B_1) \cup \cdots \cup (A_k \times B_k)$ for every possible A_i, B_i which is a union of equivalence classes of $\sim_R$. ◀

7 Discussion

In this paper, we establish among other things undecidability of the k-regular colorability problem for $k \geqslant 2$. Yet, little is known about the regular colorability problem.

▶ **Conjecture 7.1.** *The Rec-separability problem—or, equivalently, the regular colorability problem—is undecidable.*

Beyond its complexity status, the structural properties of regular colorability evades us:

▶ **Conjecture 7.2.** *Over automatic graphs, the following notions are pairwise disjoint:*

1. *to be finitely regular colorable,*
2. *to be finitely colorable,*
3. *not to contain unbounded cliques.*

Note that the implications (1) ⇒ (2) ⇒ (3) trivially hold. Moreover, recall that while the automatic tree of Example 4.5 is not 2-regular colorable, it is 3-regular colorable (it suffices to color ε with a new color, and then color $a^p b^q$ by looking at the parity of $p - q$). Hence, it does not prove that (2) $\not\Rightarrow$ (1). Likewise, on arbitrary infinite graphs, we know that there exists triangle-free graphs that are not finitely colorable [25]—but we believe these graphs not to be automatic, and hence they would not prove that (3) $\not\Rightarrow$ (2).

Finally, observe that it is decidable to test whether an automatic graph has *infinite* cliques [17, Corollary 5.5]. We conjecture that this property generalizes to unbounded cliques.

▶ **Conjecture 7.3.** *The problem of whether an automatic graph has bounded cliques is decidable.*

References

1 Pablo Barceló, Chih-Duo Hong, Xuan Bach Le, Anthony W. Lin, and Reino Niskanen. Monadic decomposability of regular relations. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 103:1–103:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.103.

2 Pablo Barceló, Leonid Libkin, Anthony Widjaja Lin, and Peter T. Wood. Expressive languages for path queries over graph-structured data. *ACM Transactions on Database Systems (TODS)*, 37(4):31, 2012. doi:10.1145/2389241.2389250.

3 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.

4 Pascal Bergsträßer, Moses Ganardi, Anthony W. Lin, and Georg Zetzsche. Ramsey quantifiers over automatic structures: Complexity and applications to verification. In Christel Baier and Dana Fisman, editors, *LICS*, pages 28:1–28:14. ACM, 2022. doi:10.1145/3531130.3533346.

5 Jean Berstel. *Transductions and Context-Free Languages*. Teubner-Verlag, 1979.

6 Achim Blumensath and Erich Grädel. Automatic structures. In *LICS*, pages 51–62. IEEE Computer Society, 2000.

7 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.

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

9 Lorenzo Clemente, Wojciech Czerwinski, Slawomir Lasota, and Charles Paperman. Regular separability of Parikh automata. In *ICALP*, pages 117:1–117:13, 2017. doi:10.4230/LIPIcs.ICALP.2017.117.

10 Wojciech Czerwinski, Wim Martens, Lorijn van Rooijen, Marc Zeitoun, and Georg Zetzsche. A characterization for decidable separability by piecewise testable languages. *Discret. Math. Theor. Comput. Sci.*, 19(4), 2017.

11 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.

12 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.

13 Hajime Ishihara, Bakhadyr Khoussainov, and Sasha Rubin. Some results on automatic structures. In *LICS*, page 235. IEEE Computer Society, 2002. doi:10.1007/11690634_22.

14 Chris Köcher. *Analyse der Entscheidbarkeit diverser Probleme in automatischen Graphen*. Unpublished manuscript, 2014. URL: https://people.mpi-sws.org/~ckoecher/files/theses/bsc-thesis.pdf.

15 Eryk Kopczynski. Invisible pushdown languages. In *LICS*, pages 867–872, 2016. doi:10.1145/2933575.2933579.

16 Dietrich Kuske and Markus Lohrey. Hamiltonicity of automatic graphs. In Giorgio Ausiello, Juhani Karhumäki, Giancarlo Mauri, and C.-H. Luke Ong, editors, *IFIP*, volume 273, pages 445–459. Springer, 2008.

17 Dietrich Kuske and Markus Lohrey. Some natural decision problems in automatic graphs. *J. Symb. Log.*, 75(2):678–710, 2010. doi:10.1007/978-0-387-09680-3_30.

18 Yves Lecerf. Machines de turing réversibles. récursive insolubilité en $n \in \mathbf{N}$ de l'équation $u = \theta^n u$, ou θ est un « isomorphisme de codes ». *Comptes Rendus hebdomadaires des seances de l'academie des sciences*, 257:2597–2600, 1963.

19 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.

20 Christof Löding and Christopher Spinrath. Decision problems for subclasses of rational relations over finite and infinite words. *Discret. Math. Theor. Comput. Sci.*, 21(3), 2019. doi:10.23638/DMTCS-21-3-4.

21 Kenichi Morita. *Reversible Turing Machines*, pages 103–156. Springer Japan, Tokyo, 2017. doi:10.1007/978-4-431-56606-9_5.

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

23 Thomas Place and Marc Zeitoun. Separating regular languages with first-order logic. *Logical Methods in Computer Science (LMCS)*, 12(1), 2016. doi:10.2168/LMCS-12(1:5)2016.

24 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.

25 Peter Ungar and Blanche Descartes. *k*-Chromatic graphs without triangles. *The American Mathematical Monthly*, 61(5):352–353, 1954. doi:10.2307/2307489.

26 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.

A Missing Proofs

▶ **Lemma 3.2.** *If R_1 and R_2 are automatic, then so is $\mathcal{I}nc_{R_1,R_2}$. Moreover, we can build an automaton for $\mathcal{I}nc_{R_1,R_2}$ in polynomial time in the size of the automata for R_1 and R_2.*

First stated in page 5.

Proof of Lemma 3.2. By definition, the incompatibility relation $\mathcal{I}nc_{R_1,R_2}$ can be written as $R_{\neg(\text{COMP}_l)} \cup R_{\neg(\text{COMP}'_l)} \cup R_{\neg(\text{COMP}_r)} \cup R_{\neg(\text{COMP}'_r)}$, where:

$$\begin{aligned}
R_{\neg(\text{COMP}_l)} &\triangleq \{(u,u') \in \mathbb{A}^* \times \mathbb{A}^* \mid \exists v \in \mathbb{A}^*,\ (u,v) \in R_1 \wedge (u',v) \in R_2\},\\
R_{\neg(\text{COMP}'_l)} &\triangleq \{(u,u') \in \mathbb{A}^* \times \mathbb{A}^* \mid \exists v \in \mathbb{A}^*,\ (u',v) \in R_1 \wedge (u,v) \in R_2\},\\
R_{\neg(\text{COMP}_r)} &\triangleq \{(u,u') \in \mathbb{A}^* \times \mathbb{A}^* \mid \exists v \in \mathbb{A}^*,\ (v,u) \in R_1 \wedge (v,u') \in R_2\}, \text{ and}\\
R_{\neg(\text{COMP}'_r)} &\triangleq \{(u,u') \in \mathbb{A}^* \times \mathbb{A}^* \mid \exists v \in \mathbb{A}^*,\ (v,u') \in R_1 \wedge (v,u) \in R_2\}
\end{aligned}$$

Observe that starting from automata for R_1 and R_2, then for each of the relation $R_{\neg(\text{COMP}_l)}$, $R_{\neg(\text{COMP}'_l)}$, $R_{\neg(\text{COMP}_r)}$ or $R_{\neg(\text{COMP}'_r)}$, we can build an automaton recognizing them using a product construction, which can be implemented in polynomial time. It then follows that we can build a polynomial automaton recognizing $\mathcal{I}nc_{R_1,R_2}$. ◀

▶ **Lemma 4.2.** *The* REACHABLE REGULARITY PROBLEM *is undecidable.*

First stated in page 7.

Proof of Lemma 4.2. By reduction from the halting problem for deterministic and reversible TMs, which is undecidable by Proposition 4.1. Given a deterministic and reversible TM T (running on the empty input), consider the TM T' where every time there is a transition $(u,p,v) \to (u',q,v')$ from configuration c to configuration c' in T, simulate this transition in T'—a's should be treated as blank symbols—, and then rewritting $a^n b^n$ into $a^{n+1}b^{n+1}$. When T writes on a blank symbol that was actually a a in T', we must also add an extra a (to account for the one that was overwritten): this case is depicted Figure 4.

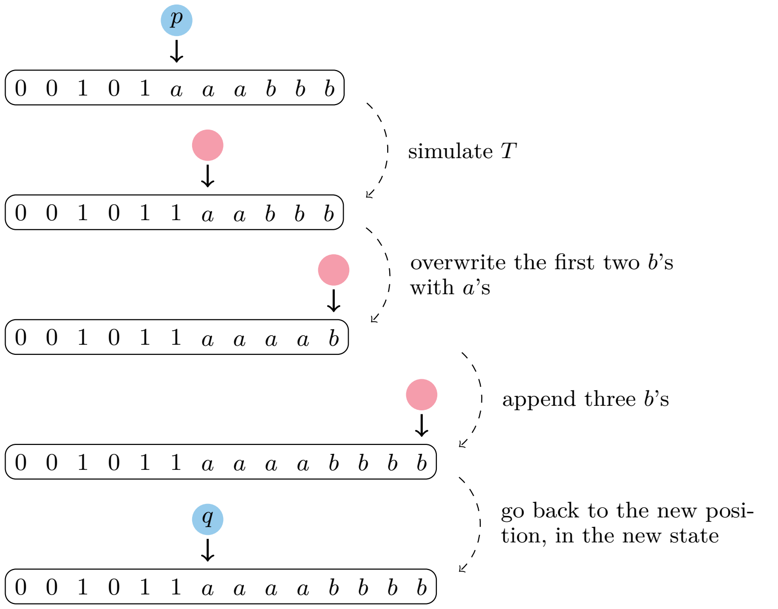

p
0 0 1 0 1 a a a b b b
simulate T
0 0 1 0 1 1 a a b b b
overwrite the first two b's with a's
0 0 1 0 1 1 a a a a b
append three b's
0 0 1 0 1 1 a a a a b b b b
go back to the new position, in the new state
q
0 0 1 0 1 1 a a a a b b b b

Figure 4 Encoding of a single transition of the form "when reading a blank in state p, write a 1, go in state q and move right" of the machine T in the machine T' in the proof of Lemma 4.2. Red unlabelled states represent states of T' that are not originally present in T.

Moreover, when T deletes a symbol at the end of the tape, we must shift the $a^n b^n$ prefix. This can be done by replacing the blank with an a, the last a with a b, and deleting the last b.

Observe that T' is a wf-RTM:

1. the initial configuration $(\bot, q_0, \bot)$ has no predecessor;
2. it is deterministic and co-deterministic:
   - every configuration inside a path $(u, q, va^n b^n) \xrightarrow{*} (u, q, va^{n+1}b^{n+1})$ has, by definition, exactly in- and out-degree one;
   - every configuration of the form $(u, p, va^n b^n)$ has as many predecessors [resp. successors] in T' as (u, q, v) in T, namely one since T was assumed to be deterministic and reversible;
3. it has no infinite descending chain since $\mathbb{N}$ is well-founded.

Moreover T' has no cycle, so if T is halting (on an empty input) then the set of reachable configurations of T' is finite (since it is a wf-RTM) and thus regular. If T is not halting, the set of reachable configurations of T' is infinite and its projection onto $\{a, b\}$ is an infinite set of words of the form $a^n b^{n'}$ where $n - 2 \leqslant n' \leqslant n + 2$. Hence, since regular languages are closed under homomorphic images, the reachable configurations of T' cannot be regular. ◀

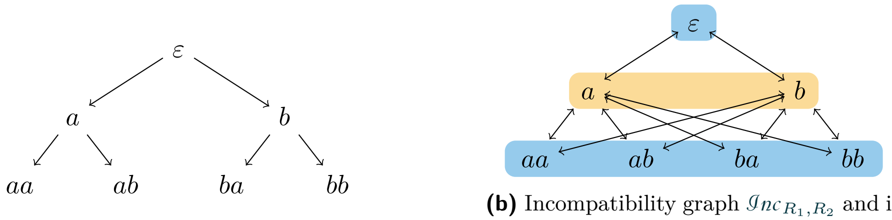
B Incompatibility Graph

▶ **Example B.1.** Let $\mathbb{A} = \{a, b\}$, R_1 be the equal-length relation, and

$$R_2 = \{(u, ua) \mid u \in \mathbb{A}^*\} \cup \{(u, ub) \mid u \in \mathbb{A}^*\}.$$

Then, u is incompatible with u' if $|u| = |u'| + 1$ (this is given by $(\textsc{comp}_l)$ or $(\textsc{comp}_r)$), or if $|u'| = |u| + 1$ (this is given by $(\textsc{comp}'_l)$ or $(\textsc{comp}'_r)$). R_2 and the incompatibility graph are depicted in Figure 5.

Note that R_1 and R_2 are separable by the recognizable relation S consisting of all pairs (u, v) such that $|u|$ and $|v|$ have the same parity. Moreover, $\mathcal{I}nc_{R_1,R_2}$ is 2-regular colorable, the two colors being the words of even and odd length.



(a) The relation R_2.

(b) Incompatibility graph $\mathcal{I}nc_{R_1,R_2}$ and its 2-regular coloring.

Figure 5 Automatic graphs of Example B.1, restricted to words of length at most 2.