Boolean Combinations of ω-Rational Trace Languages: Emptiness, Rationality, Regularity

Dietrich Kuske
Technische Universität Ilmenau, Germany

Abstract

This paper studies decision problems for Boolean combinations of ω-rational trace languages. The complexities of these decision problems (emptiness, regularity, rationality) are classified depending on properties of the independence alphabets and of the Boolean combinations allowed. For any of the problems, we obtain a trichotomy ranging from decidable over a low level of the arithmetical hierarchy to a low level of the analytical hierarchy.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Theory of computation → Regular languages

Keywords and phrases Mazurkiewicz traces, rational and regular languages, degrees of unsolvability

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.17

1 Introduction

Finite and infinite Mazurkiewicz traces [25, 8] (cf. [7]) are a formal model for the description of the (nonterminating) behavior of concurrent systems that synchronize via shared actions. They are closely related to communicating finite state machines [5, 12, 13, 23], have been studied in the contexts of model checking (e.g., [10, 3]) and the distributed control synthesis problem [11, 14, 15] and allow modeling multi-buffer systems [18] and multi-pushdown systems [19, 20, 22].

The idea[1] behind traces is that sequences of basic actions, i.e., words, form interleavings of the executions of several, concurrently running, processes. Words that are interleavings of one and the same concurrent execution are considered equivalent. Formally, the concurrency in a system is given by an independence alphabet that describes which pairs of actions are executed by disjoint sets of sequential processes and can therefore commute in interleaving executions. As a consequence, questions like "Do two concurrent systems allow the same concurrent executions?" translate into "Does every interleaving execution of one system have an equivalent interleaving execution of the other system?" or "Are the closures of the sets of interleaving executions under the equivalence of words equal?" In other words, one is interested in the closure of a language under this equivalence.

For regular sets of finite words, questions of this type have been studied in detail. Here, e.g., non-inclusion asks for the existence of some finite object (a word) with some decidable property (to belong to the closure of the first regular set, but not to the closure of the second). Hence inclusion belongs to the first universal level of the arithmetical hierarchy Π^0_1. Differently, regularity asks for the existence of a finite object (a finite automaton) such that all finite objects (words) satisfy some decidable property (to belong to the closure if, and only if, they are accepted by the automaton). Hence, this problem belongs to the second existential level of the arithmetical hierarchy Σ^0_2. These easy observations do by no means exclude that inclusion or regularity can be expressed by simpler means, e.g., are decidable or belong to Π^0_1.

[1] These two paragraphs are from [21].

51st International Symposium on Mathematical Foundations of Computer Science (MFCS 2026).
Editors: Michal Koucký and Daniela Petrişan; Article No. 17; pp. 17:1–17:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The computer science literature provides a clear distinction between the decidable and undecidable instances of these problems. This border is described using two properties of independence alphabets, namely "transitivity" and "diagonality" (which is strictly more general). The known answers are as follows: inclusion and regularity are decidable for transitive independence alphabets, and otherwise complete for Π^0_1 and Σ^0_2, respectively [2, 28, 21]; disjointness is decidable even for diagonal independence alphabets and Π^0_1-complete otherwise [1].

An attempt to extend these results to infinite traces was started in [21]. That paper shows in particular that, for closures of regular languages of infinite traces, regularity for transitive independence alphabets and disjointness for diagonal independence alphabets are in Π^0_1; it was left open as to whether these problems are decidable as they are in the finite case [28, 1]. The results of this paper imply affirmative answers to both these questions.

In detail, we study decision problems for Boolean combinations of closures of regular ω-languages, namely the *emptiness*, the *regularity*, and the *rationality* problem that asks whether a given Boolean combination is the closure of some regular language (this latter problem has been considered for finite words in [2], but does not appear in [21]). For infinite words, all these problems belong to the second universal level of the analytical hierarchy Π^1_2. Depending on graph theoretic properties of the independence alphabet (i.e., the "architecture" of the distributed system) and properties of the Boolean combination, we classify all these problems into decidable, Π^0_1-complete, Σ^0_2-complete, and Π^1_1-hard (cf. Table 1). The only graph properties appearing in this classification are "transitive" and "diagonal"; and the only properties of the Boolean combinations are "trivial" (i.e., $\emptyset$ or "all words"), "finite union", and "positive". Looking at the results, one observes two striking similarities between the cases of finite and infinite words:

1. A problem for infinite words is decidable if, and only if, its version for finite words is decidable.
2. If one considers only diagonal independence alphabets, then there is no difference in complexity between a problem for infinite words and for finite words.

One also notes that the regularity problem and the rationality problem behave much the same: the levels of undecidability are almost always the same. Nevertheless, these two problems are quite different: the regularity problem is of maximal difficulty even for the trace closure of a rational language (where the rationality problem has a trivial answer).

On the other hand, there are also two striking differences:

1. Any of the problems on finite words yields a dichotomy while, for infinite words, we always have a trichotomy.
2. If one allows some non-diagonal independence alphabet, then the problem for finite words remains arithmetical while it is Π^1_1-hard for infinite words.

2 Preliminaries

The the set $\mathbb{N}$ of non-negative integers starts with 0 and, for $n \in \mathbb{N}$, we set $[n] = \{1, \ldots, n\}$. The powerset of a set X is denoted $\mathcal{P}(X)$.

Words and automata

Let Γ be an alphabet. The set of finite words over Γ is denoted Γ^*, Γ^ω denotes the set of ω-words over Γ, and $\Gamma^\infty = \Gamma^* \cup \Gamma^\omega$. For two words or ω-words $x, y \in \Gamma^\infty$, the *shuffle* $x \sqcup\!\sqcup\, y \subseteq \Gamma^\infty$ is the set of all words or ω-words $x_0y_0\, x_1y_1\, x_2y_2 \cdots$ where $x_i, y_i \in \Gamma^*$ are finite words with $x = x_0x_1x_2 \cdots$ and $y = y_0y_1y_2 \cdots$. For $\alpha \in \Gamma^\infty$ and $A \subseteq \Gamma$, the *projection*

Table 1 Summary of results.
Here, $\mathbb{A} \subseteq \mathrm{IndA}$ and $\Phi \subseteq \mathrm{BF}$ are decidable sets. In the second column, one finds the conditions on $\mathbb{A}$ and Φ that make the problem decidable. The last column describes the complexity in case $\mathbb{A}$ and Φ do not satisfy the condition in the second column. For ω-words, the last column contains several complexities together with the conditions on $\mathbb{A}$ and Φ that characterize this complexity ("(Π^1_1, Π^1_2)" means "Π^1_1-hard and in Π^1_2").

problem	decidable	otherwise
$\mathrm{EmptP}_*(\mathbb{A}, \Phi)$ (Theorem 3.2)	$\mathbb{A} \subseteq \mathrm{TransA}$ or $\mathbb{A} \subseteq \mathrm{DiagA}$ and $\Phi \subseteq \mathrm{monBF}$ or $\Phi \subseteq \mathrm{MAX}$	Π^0_1-complete
$\mathrm{EmptP}_\omega(\mathbb{A}, \Phi)$	$\mathbb{A} \subseteq \mathrm{TransA}$ (Thm. 4.1) or $\mathbb{A} \subseteq \mathrm{DiagA}$ and $\Phi \subseteq \mathrm{monBF}$ (Thm. 5.2) or $\Phi \subseteq \mathrm{MAX}$ (Prop. 2.3)	Π^0_1-complete (Thm. 3.2 and Cor. 5.3) (if $\mathbb{A} \not\subseteq \mathrm{TransA}$, $\mathbb{A} \subseteq \mathrm{DiagA}$ and $\Phi \not\subseteq \mathrm{monBF}$) or Π^1_1-complete (Thm. 6.2) (if $\mathbb{A} \not\subseteq \mathrm{DiagA}$, $\Phi \not\subseteq \mathrm{MAX}$, $\Phi \subseteq \mathrm{monBF}$) or (Π^1_1, Π^1_2) (Thm. 6.2) (if $\mathbb{A} \not\subseteq \mathrm{DiagA}$ and $\Phi \not\subseteq \mathrm{monBF}$)
$\mathrm{RatP}_*(\mathbb{A}, \Phi)$ (Theorem 3.4)	$\mathbb{A} \subseteq \mathrm{TransA}$ or $\Phi \subseteq \mathrm{MAX}$	Σ^0_2-complete
$\mathrm{RatP}_\omega(\mathbb{A}, \Phi)$	$\mathbb{A} \subseteq \mathrm{TransA}$ (Thm. 4.1) or $\Phi \subseteq \mathrm{MAX}$ (Prop. 2.3)	Σ^0_2-complete (Thm. 5.4 and 3.4(2)) (if $\mathbb{A} \not\subseteq \mathrm{TransA}$, $\mathbb{A} \subseteq \mathrm{DiagA}$, $\Phi \not\subseteq \mathrm{MAX}$) or (Π^1_1, Π^1_2) (Thm. 6.2(2)) (if $\mathbb{A} \not\subseteq \mathrm{DiagA}$ and $\Phi \not\subseteq \mathrm{MAX}$)
$\mathrm{RegP}_*(\mathbb{A}, \Phi)$ (Theorem 3.4)	$\mathbb{A} \subseteq \mathrm{TransA}$ or $\Phi \subseteq \mathrm{CONST}$	Σ^0_2-complete
$\mathrm{RegP}_\omega(\mathbb{A}, \Phi)$	$\mathbb{A} \subseteq \mathrm{TransA}$ (Thm. 4.3) or $\Phi \subseteq \mathrm{CONST}$ (Prop. 2.3)	Σ^0_2-complete (Thm 5.4 and 3.4(2')) (if $\mathbb{A} \not\subseteq \mathrm{TransA}$, $\mathbb{A} \subseteq \mathrm{DiagA}$, $\Phi \not\subseteq \mathrm{CONST}$) or (Π^1_1, Π^1_2) (Thm. 6.2) (if $\mathbb{A} \not\subseteq \mathrm{TransA}$ and $\Phi \not\subseteq \mathrm{CONST}$)

$\mathrm{proj}_A(\alpha) \in A^\infty$ is obtained by deleting all letters from α that do not belong to A. We write $|\alpha|_A$ for the length of the projection $\mathrm{proj}_A(\alpha)$; $\mathrm{proj}_a(\alpha)$, $|\alpha|_a$, and $|\alpha|$ abbreviate $\mathrm{proj}_{\{a\}}(\alpha)$, $|\alpha|_{\{a\}}$, and $|\alpha|_\Gamma$, respectively.

Subsets of Γ^* are called *finitary languages* or $*$*-languages* while subsets of Γ^ω are ω-*languages*.

Let $K \subseteq \Gamma^*$ and $L \subseteq \Gamma^\infty$. Then the *concatenation* of K and L is $K \cdot L = \{u\,x \mid u \in K \text{ and } x \in L\} \subseteq \Gamma^\infty$. The *Kleene iteration* K^* is the set of words $u_1\,u_2 \cdots u_m$ for some $m \geq 0$ with $u_1, u_2, \ldots, u_m \in K$. The ω*-iteration* K^ω of K is the set of ω-words $u_1\,u_2 \cdots$ with $u_i \in K \setminus \{\varepsilon\}$ for all $i \geq 1$.

Let $K, L \subseteq \Gamma^\infty$. The *shuffle* of K and L is the union of all shuffles $x \sqcup\!\sqcup\, y$ with $x \in K$ and $y \in L$.

A *nondeterministic finite automaton* or *nfa* is a tuple $\mathcal{A} = (Q, \Gamma, S, T, F)$ where Q is a finite set of "states", Γ an alphabet, $S, F \subseteq Q$ sets of "initial" and "final states", respectively, and $T \subseteq Q \times \Gamma \times Q$ the transition relation. For a state $p \in Q$, we write $\mathcal{A}^{p\rightarrow}$ for the nfa $(Q, \Gamma, \{p\}, T, F)$ that only differs from $\mathcal{A}$ in its initial state. For $p, q \in Q$ and $u \in \Gamma^*$, we write $p \xrightarrow{u} q$ if there is some u-labeled path from p to q. If there is such a path that visits some accepting state, we write $p \overset{u}{\Longrightarrow} q$. A word $u \in \Gamma^*$ is *accepted* by $\mathcal{A}$ if there are $p \in S$ and $q \in F$ with $p \xrightarrow{u} q$; $L^*(\mathcal{A})$ denotes the $*$-language of words accepted by $\mathcal{A}$. A $*$-language $L \subseteq \Gamma^*$ is *regular* if there exist an nfa $\mathcal{A}$ with $L = L^*(\mathcal{A})$.

An ω-word α is *accepted* by $\mathcal{A}$ if it splits into finite words $\alpha = u_0\,u_1 \cdots$ such that $p \xrightarrow{u_0} q$ and $q \xrightarrow{u_i} q$ for all $i \geq 1$ for some $p \in S$ and $q \in F$. The set of ω-words accepted by $\mathcal{A}$ is $L^\omega(\mathcal{A})$; an ω-language L is *regular* if there is some nfa $\mathcal{A}$ with $L = L^\omega(\mathcal{A})$.

Finally, we recall some classical notions on binary relations on Γ^∞ from [16], cf. [4] for the case of relations on finite words.

Let $R \subseteq \Gamma^* \times \Gamma^*$ and $S \subseteq \Gamma^\infty \times \Gamma^\infty$ be two binary relations. Then $R \cdot S = \{(uv, u'v') \mid (u, u') \in R \text{ and } (v, v') \in S\}$ is the *concatenation* of R and S. The *power* of R is defined inductively by $R^0 = \{(\varepsilon, \varepsilon)\}$ and $R^{n+1} = R \cdot R^n$. The *finite iteration* of R is $R^* = \bigcup_{n \geq 0} R^n$; its ω*-iteration* is the set

$$R^\omega = \{(u_1 u_2 u_3 \cdots, v_1 v_2 v_3 \cdots) \in \Gamma^\infty \times \Gamma^\infty \mid (u_i, v_i) \in R \text{ for all } i \geq 1\}\,.$$

The relation $S \subseteq \Gamma^\infty \times \Gamma^\infty$ is *rational* if it can be constructed from finite relations on Γ^* using union, concatenation (with the first factor a relation on Γ^*), and finite and ω-iteration (applied to relations on Γ^*).

Rational relations describe the behavior of transducers: a *transducer* is a tuple $\mathcal{A} = (Q, \Gamma, S, T, F)$ where Q, Γ, S and F are as in nfas and the transition relation T is a subset of $Q \times (\Gamma \cup \{\varepsilon\})^2 \times Q$. For $p, q \in Q$ and $u, v \in \Gamma^*$, we write $p \xrightarrow{(u,v)} q$ if there is some (u, v)-labeled path from p to q. A pair of possibly infinite words $(\alpha, \beta) \in \Gamma^\infty \times \Gamma^\infty$ is *accepted* by $\mathcal{A}$ if α and β split into $\alpha = u_0\,u_1 \cdots$ and $\beta = v_0\,v_1 \cdots$ such that there are states $p \in S$ and $q \in F$ with $p \xrightarrow{(u_0,v_0)} q$ and $q \xrightarrow{(u_i,v_i)} q$ for all $i \geq 1$ (if $u_i = v_i = \varepsilon$ for all $i > j$, then α and β both are finite and there is a sequence of transitions from q back to q that all are labeled $(\varepsilon, \varepsilon)$). The set of pairs accepted by $\mathcal{A}$ is $R^\infty(\mathcal{A})$. Then a relation on Γ^∞ is rational iff it is accepted by some transducer.

From an nfa $\mathcal{A}$ and a transducer $\mathcal{B}$, one can construct nfas $\mathcal{A}_*$ and $\mathcal{A}_\omega$ with $v \in L^*(\mathcal{A}_*)$ iff there exists $u \in L^*(\mathcal{A})$ with $(u, v) \in R^\infty(\mathcal{B})$ and $\beta \in L^\omega(\mathcal{A}_\omega)$ iff there exists $\alpha \in L^\omega(\mathcal{A})$ with $(\alpha, \beta) \in R^\infty(\mathcal{B})$.

▶ **Convention.** Whenever, for an nfa $\mathcal{A}$, we are mainly interested in its ω-language $L^\omega(\mathcal{A})$, we will speak of $\mathcal{A}$ as a *Büchi-automaton*.

Many objects considered in this paper appear in two forms: word vs. ω-word, *-language vs. ω-language. At places, we blur this distinction and just talk about words and languages assuming that the context explains what version is meant.

Traces

An *independence alphabet* is a pair (Γ, I) of an alphabet Γ and an irreflexive and symmetric *independence relation* $I \subseteq \Gamma \times \Gamma$; its complement is the *dependence relation* $D = \Gamma^2 \setminus I$. The set of all independence alphabets is denoted IndA.

Two words $x, y \in \Gamma^\infty$ are *equivalent* [8] (denoted $x \sim y$) if, for all (not necessarily distinct) letters $a, b \in \Gamma$ with $(a, b) \in D$, we have $\mathrm{proj}_{\{a,b\}}(x) = \mathrm{proj}_{\{a,b\}}(y)$.[2]

For $L \subseteq \Gamma^\infty$, the *closure* is the set $[L] = \{y \in \Gamma^\infty \mid \exists x \in L\colon x \sim y\}$ of all (ω-)words y that are equivalent to some element of L. The set $L \subseteq \Gamma^\infty$ is *closed* if $L = [L]$. If $L = \{x\}$ is a singleton, the closure $[L]$ of L equals the equivalence class $[x]$ of x wrt. $\sim$.

In this paper, we will regularly be concerned with closures of regular sets and with closed regular sets. If L is closed and regular, then it is the closure $[L]$ of the regular set L. Conversely, suppose $(a, b) \in I$. The closure of the regular set $L = \{ab\}^*$ is the set $[L]$ of words $u \in \{a, b\}^*$ with $|u|_a = |u|_b$ and therefore not regular. Thus, regular closed sets are closures of regular sets, but there are closures of regular sets that are not regular.

▶ Remark. Recognizable and rational sets of (infinite) traces [25, 9] correspond to closed regular and closures of regular languages, respectively. We could therefore phrase our results in terms of these sets of traces; the author prefers the current point of view as it only concerns sets of words and avoids the consideration of traces as equivalence classes of words.

Decision problems

This paper deals with decision problems related to Boolean combinations of closures of regular languages, e.g., we will investigate the decidability status of the question whether the language $[L^*(\mathcal{A}_1)] \cap [L^*(\mathcal{A}_2)]$ is empty (is regular, resp.) for two nfas $\mathcal{A}_1$ and $\mathcal{A}_2$. To present all these problems uniformly, we next introduce some notions.

▶ **Definition 2.1.** *Let $\odot \in \{*, \omega\}$, $\varphi\colon \{0,1\}^n \to \{0,1\}$ a Boolean function, and $L_1, \ldots, L_n \subseteq \Gamma^\odot$ $\odot$-languages over Γ. The $\odot$-language $\big\langle L_1, \ldots, L_n \big\rangle^\varphi$ is the set of words $x \in \Gamma^\odot$ such that $\varphi\big(\chi_{L_1}(x), \chi_{L_2}(x), \ldots, \chi_{L_n}(x)\big) = 1$ where χ_{L_i} is the characteristic function of L_i.*

▶ **Example.** Let Id be the unary identity function, $\wedge$ the binary minimum-function, and $\neg$ the negation. Then, for any languages L_1 and L_2, we have $\big\langle L_1 \big\rangle^{\mathrm{Id}} = L_1$, $\big\langle L_1, L_2 \big\rangle^{\wedge} = L_1 \cap L_2$, and $\big\langle L_1 \big\rangle^{\neg} = \Gamma^\odot \setminus L_1$.

We now come to the decision problems studied in this paper.

▶ **Definition 2.2.** *Let $\odot \in \{*, \omega\}$, $\mathbb{A} \subseteq \mathrm{IndA}$ be some set of independence alphabets, Φ some set of Boolean functions, and $\mathrm{GenP} \in \{\mathrm{EmptP}, \mathrm{RatP}, \mathrm{RegP}\}$. Then $\mathrm{GenP}_\odot(\mathbb{A}, \Phi)$ is the set of tuples $\big((\Gamma, I), \varphi, \mathcal{A}_1, \ldots, \mathcal{A}_n\big)$ with $(\Gamma, I) \in \mathbb{A}$ an independence alphabet from $\mathbb{A}$, $\varphi \in \Phi$ an n-ary Boolean function from Φ, and $\mathcal{A}_1, \ldots, \mathcal{A}_n$ nfas over Γ with*

$$H = \Big\langle [L^\odot(\mathcal{A}_1)], [L^\odot(\mathcal{A}_2)], \ldots, [L^\odot(\mathcal{A}_n)] \Big\rangle^\varphi$$

such that

[2] Two other, equivalent, definitions of $\sim$ have been formulated by Gastin [8].

- $\mathrm{GenP} = \mathrm{EmptP}$ *and* $H = \emptyset$ *(the* emptiness problem*),*
- $\mathrm{GenP} = \mathrm{RatP}$ *and* $H = [L]$ *for some regular language* $L \subseteq \Gamma^{\odot}$ *(the* rationality problem*),*
- *or* $\mathrm{GenP} = \mathrm{RegP}$ *and* H *is regular (the* regularity problem*).*

▶ **Example.** For non-diagonal independence alphabets (Γ, I), the problems $\mathrm{EmptP}_*((\Gamma, I), \wedge)$ (the "disjointness problem") and $\mathrm{RegP}_*((\Gamma, I), \mathrm{Id})$ are undecidable [1, 28].

The above problems are parametrized by a set $\mathbb{A}$ of independence alphabets and a set Φ of Boolean functions. We next describe particular sets that are central in our classification.

A Boolean function $\varphi\colon \{0,1\}^n \to \{0,1\}$ is

- *constant* if $\varphi(\overline{b}) = \varphi(\overline{c})$ holds for all $\overline{b}, \overline{c} \in \{0,1\}^n$.
- a *max-function* if it is constant or there exists a non-empty set $A \subseteq [n]$ with $\varphi(b_1, \ldots, b_n) = \max\{b_i \mid i \in A\}$ for all $b_1, \ldots, b_n \in \{0,1\}$.
- *monotone* if $\varphi(b_1, \ldots, b_n) \leq \varphi(c_1, \ldots, c_n)$ for any arguments from $\{0,1\}$ with $b_i \leq c_i$ for all $i \in [n]$.

The set of all constant (max-, monotone, resp.) functions is denoted CONST (MAX, monBF, resp.); BF denotes the set of all Boolean functions.

It turns out that the regularity problem for constant functions and the emptiness problem and the rationality problem for max-functions are simple (if not trivial).

▶ **Proposition 2.3.** *Let* $\odot \in \{*, \omega\}$*. Then* $\mathrm{EmptP}_\odot(\mathrm{IndA}, \mathrm{MAX})$*,* $\mathrm{RatP}_\odot(\mathrm{IndA}, \mathrm{MAX})$*, and* $\mathrm{RegP}_\odot(\mathrm{IndA}, \mathrm{CONST})$ *are decidable.*

After considering constant and max-functions, we next turn attention to Boolean functions outside these classes. Claim (1) in the following lemma ensures that the function Id induces the full difficulty of the regularity problem $\mathrm{RegP}_\odot(\mathbb{A}, \Phi)$ if $\Phi \not\subseteq \mathrm{CONST}$. Similarly, $\wedge \in \mathrm{monBF} \setminus \mathrm{MAX}$ ($\neg \notin \mathrm{monBF}$, resp.) gives rise to the full difficulty of $\mathrm{GenP}_\omega(\mathbb{A}, \Phi)$ if $\Phi \subseteq \mathrm{monBF}$ and $\Phi \not\subseteq \mathrm{MAX}$ ($\Phi \not\subseteq \mathrm{monBF}$).

▶ **Lemma 2.4.** *Let* $\odot \in \{*, \omega\}$, $\mathrm{GenP} \in \{\mathrm{EmptP}, \mathrm{RatP}, \mathrm{RegP}\}$*,* $\mathbb{A} \subseteq \mathrm{IndA}$*, and* $\varphi \in \mathrm{BF}$.

(1) *If* $\varphi \notin \mathrm{CONST}$*, then* $\mathrm{RegP}_\odot(\mathbb{A}, \mathrm{Id}) \leq \mathrm{RegP}_\odot(\mathbb{A}, \varphi)$.

(2) *If* $\varphi \in \mathrm{monBF} \setminus \mathrm{MAX}$*, then* $\mathrm{GenP}_\odot(\mathbb{A}, \wedge) \leq \mathrm{GenP}_\odot(\mathbb{A}, \varphi)$.

(3) *If* $\varphi \notin \mathrm{monBF}$*, then* $\mathrm{GenP}_\odot(\mathbb{A}, \neg) \leq \mathrm{GenP}_\odot(\mathbb{A}, \varphi)$.

Besides the above classes of Boolean functions, there are also classes of independence alphabets central in this paper. Let (Γ, I) be an independence alphabet. It is

- *complete* if $(a, b) \in I$ for all $a, b \in \Gamma$ distinct,
- *transitive* if P_3 (the path on three vertices) is not an induced subgraph, and
- *diagonal* if neither C_4 (the cycle on four vertices) nor P_4 is an induced subgraph.

The term "diagonal" was coined in [29], [1] uses the term "transitive forest" instead. The set of all transitive (diagonal, resp.) independence alphabets is denoted TransA (DiagA, respectively).

Arithmetical and analytical hierarchy

This paper describes the complexity of the above decision problems. But as most of them are undecidable, this requires a yardstick for undecidability. The said hierarchies are such a yardstick. We refer the reader to [27] for definitions and useful properties.

3 Finitary languages

In this section, we study the problems $\mathrm{GenP}_*(\mathbb{A},\Phi)$, i.e., consider finite words. Many of the results follow easily from the literature, an exception is the rationality problem for non-transitive independence alphabets.

We *first state* upper bounds on some versions of the emptiness problem $\mathrm{EmptP}_*(\mathbb{A},\Phi)$, thanks to the work by Aalbersberg, Welzl, and Hoogeboom, the proofs of claim (2) and (3) are rather straightforward. Claim (1) follows since containment in the closure of a regular language is decidable.

▶ **Lemma 3.1.**
(1) *The emptiness problem* $\mathrm{EmptP}_*(\mathrm{IndA},\mathrm{BF})$ *is in* Π^0_1.
(2) *The emptiness problem* $\mathrm{EmptP}_*(\mathrm{TransA},\mathrm{BF})$ *is decidable [2].*
(3) *The emptiness problem* $\mathrm{EmptP}_*(\mathrm{DiagA},\mathrm{monBF})$ *is decidable [1].*

Now the following dichotomy result follows easily from [1, 26].

▶ **Theorem 3.2.** *Let* $\mathbb{A}\subseteq\mathrm{IndA}$ *and* $\Phi\subseteq\mathrm{BF}$ *be decidable sets.*
(1) *The emptiness problem* $\mathrm{EmptP}_*(\mathbb{A},\Phi)$ *is decidable if* $\mathbb{A}\subseteq\mathrm{TransA}$*, or* $\mathbb{A}\subseteq\mathrm{DiagA}$ *and* $\Phi\subseteq\mathrm{monBF}$*, or* $\Phi\subseteq\mathrm{MAX}$.
(2) *Otherwise, the emptiness problem* $\mathrm{EmptP}_*(\mathbb{A},\Phi)$ *is* Π^0_1*-hard.*

We next turn to the rationality problem, i.e., the question whether a given Boolean combination of closures of regular languages is the closure of some regular language.

We start by presenting two upper bounds. Again, citing work by Aalbersberg and Welzl, the first claim follows easily, and also the second follows easily from the definition of the rationality problem.

▶ **Lemma 3.3.**
(1) *The rationality problem* $\mathrm{RatP}_*(\mathrm{TransA},\mathrm{BF})$ *is decidable [2].*
(2) *The rationality problem* $\mathrm{RatP}_*(\mathrm{IndA},\mathrm{BF})$ *is in* Σ^0_2.

For a lower bound, one first proves the Σ^0_2-hardness of $\mathrm{RatP}_*((\Gamma,I),\varphi)$ for any non-transitive independence alphabet (Γ,I) and any of the two Boolean functions $\wedge$ and $\neg$. This is achieved in two steps (where $\varphi\in\{\wedge,\neg\}$):

The first step proves that $\mathrm{RatP}_*(\mathrm{StarA},\varphi)$ is Σ^0_2-hard where StarA is the set of all independence alphabets (Σ,I) that form a star.

To show this, let $\alpha,\beta\colon A^+\to B^+$ be homomorphisms, $L\subseteq A^+$ regular, $c\notin A\cup B$, and (Σ,I) the star with center c and leaves $A\cup B$. Furthermore, let $K=\{u\in L\mid \alpha(u)=\beta(u)\}$. For $\gamma\in\{\alpha,\beta\}$, let W_γ denote the union of all languages $u\,\gamma(u)\sqcup c^{|\gamma(u)|}$ for $u\in L$. One first shows that K is finite iff $W_\alpha\cap W_\beta$ is the closure of some regular language iff $[L\cdot(Bc)^*]\setminus(W_\alpha\cap W_\beta)$ is the closure of some regular language. Following [24, 26], one can construct nfas $\mathcal{A}$ and $\mathcal{B}$ with $\Sigma^*\setminus[L^*(\mathcal{A})]=W_\alpha\cap W_\beta$ and $L^*(\mathcal{B})=L\cdot(Bc)^*$. It follows that K is finite iff $((\Sigma,I),\neg,\mathcal{A})\in\mathrm{RatP}_*(\mathrm{StarA},\neg)$ iff $((\Sigma,I),\wedge,\mathcal{A},\mathcal{B})\in\mathrm{RatP}_*(\mathrm{StarA},\wedge)$. Since, by [21], the finiteness of K is Σ^0_2-hard, the claim follows.

In the *second step*, one proves $\mathrm{RatP}_*(\mathrm{StarA},\varphi)\le\mathrm{RatP}_*((\Gamma,I),\varphi)$ for any non-transitive independence alphabet (Γ,I).

So let $a,b,c\in\Gamma$ be three letters with $(a,c),(b,c)\in I$ and $(a,b)\in D$. And let (Σ,I) be a star with center c and leaves $a_1,\dots,a_n$. Then define the homomorphism $f\colon\Sigma^*\to\Gamma^*$ by $f(c)=c$ and $f(a_i)=ba^i$ for all $i\in[n]$. Then, for any $u,v\in\Sigma^*$, one obtains $u\sim v\iff f(u)\sim f(v)$. From this, it follows that $[K]=[L_1]\cap[L_2]\iff[f(K)]=[f(L_1)]\cap[f(L_2)]$ as

well as $[K] = \Sigma^* \setminus [L_1] \iff [f(K)] = [f(\Sigma^*)] \setminus [f(L_1)]$ for any languages $K, L_1, L_2 \subseteq \Sigma^*$. For the reduction, one also needs that $[L_1] \cap [L_2]$ is the closure of some regular language K iff this holds for $[f(L_1)] \cap [f(L_2)]$ (and similarly for $\Sigma^* \setminus [L_1]$ and $\Gamma^* \setminus [f(L_1)]$). While the implication "$\Rightarrow$" is straightforward considering $f(K)$, the converse is more involved since, e.g., there can be a regular language $K' \subseteq \Gamma^*$ with $[K'] = [f(L_1)] \cap [f(L_2)]$ and $K' \cap f(\Sigma^*) = \emptyset$. Considering the pre-image of K' under f yields the empty set whose closure need not be the intersection $[L_1] \cap [L_2]$. To prove the implication "$\Leftarrow$", one first constructs a partial rational function $r\colon \Gamma^* \to \Gamma^*$ on Γ that "computes" for every word $U \in [f(\Sigma^*)]$ a "witness" $V \in f(\Sigma^*)$ with $U \sim V$, i.e., satisfies $U \sim r(U) \in f(\Sigma^*)$ whenever $r(U)$ is defined which is the case for any $U \in [f(\Sigma^*)]$. Then the closure of the regular language $f^{-1}(r(K'))$ equals $[L_1] \cap [L_2]$ and $\Sigma^* \setminus [L_1]$, respectively. As a result, one obtains that also the rationality problems $\mathrm{RatP}_*((\Gamma, I), \wedge)$ and $\mathrm{RatP}_*((\Gamma, I), \neg)$ are Σ^0_2-hard.
In summary, we get the following dichotomy that covers arbitrary sets of independence alphabets and Boolean functions. The lower bound holds due to Lemma 2.4(2,3) and the above arguments.

For the regularity problem, the lower bounds can be shown using the arguments for [21, Thm. 3.1] (that build on [28] and the above version of Post's correspondence problem); the upper bounds follow similarly to those of the rationality problem as well as from [2, 28].

▶ **Theorem 3.4.** *Let $\mathbb{A} \subseteq \mathrm{IndA}$ and $\Phi \subseteq \mathrm{BF}$ be decidable sets.*

(1) *If $\mathbb{A} \subseteq \mathrm{TransA}$ or $\Phi \subseteq \mathrm{MAX}$, then the rationality problem $\mathrm{RatP}_*(\mathbb{A}, \Phi)$ is decidable.*

(2) *Otherwise, the rationality problem $\mathrm{RatP}_*(\mathbb{A}, \Phi)$ is Σ^0_2-complete.*

(1') *If $\mathbb{A} \subseteq \mathrm{TransA}$ or $\Phi \subseteq \mathrm{CONST}$, then the regularity problem $\mathrm{RegP}_*(\mathbb{A}, \Phi)$ is decidable.*

(2') *Otherwise, the regularity problem $\mathrm{RegP}_*(\mathbb{A}, \Phi)$ is Σ^0_2-complete.*

4 Transitive independence alphabets

From now on, we only consider the ω-versions of the decision problems.

By [21, Thm. 4.5], the "universality problem" $\mathrm{EmptP}_\omega(\mathrm{TransA}, \neg)$ and the "disjointness problem" $\mathrm{EmptP}_\omega(\mathrm{TransA}, \wedge)$ both are decidable for transitive independence alphabets. Furthermore, the regularity problem $\mathrm{RegP}_\omega(\mathrm{TransA}, \mathrm{Id})$ is in Σ^0_1 [21, Thm. 4.6]. In this section, we improve on this latter result by demonstrating its decidability. This proof is based on our first new result, namely that any Boolean combination of closures of regular languages is the closure of a regular language (assuming the independence alphabet to be transitive). From these two results, we finally infer that all problems $\mathrm{GenP}_\omega(\mathrm{TransA}, \mathrm{BF})$ are decidable.

But before we can do so, we first recall and extend the main tool of the two decidability proofs, namely types (called "characteristics" in [21]) and type automata. To this aim, let $(\Gamma, I) \in \mathrm{TransA}$ be the disjoint union of the complete independence alphabets (Γ_k, I_k) for $k \in [m]$, let φ be an n-ary Boolean function, and let $\mathcal{A}_i = (Q_i, \Gamma, S_i, T_i, F_i)$ for $i \in [n]$ be Büchi-automata over Γ. Words from $\bigcup_{k \in [m]} \Gamma_k^\infty \setminus \{\varepsilon\}$ are called *mono-alphabetic* since they use letters from only one of the subalphabets Γ_k.

For a finite mono-alphabetic word $u \in \Gamma_k^+$ with $k \in [m]$, define the *type* $\mathrm{tp}_+(u) = \big((R_i, R_i^F)_{i \in [n]}, k\big)$ setting, for $i \in [n]$, $R_i = \{(p, q) \in Q_i^2 \mid \exists u' \sim u\colon p \xrightarrow{u'}_{\mathcal{A}_i} q\}$ and $R_i^F = \{(p, q) \in Q_i^2 \mid \exists u' \sim u\colon p \xRightarrow{u'}_{\mathcal{A}_i} q\}$ [21]. Note that, differently from classical transition profiles (cf. [6]), we collect all pairs of states that are connected by a path whose label is some word from $[u]$ that might be different from u. Here, we also define the type of an infinite mono-alphabetic word $\alpha \in \Gamma_k^\omega$: it is $\mathrm{tp}_\omega(\alpha) = \big((A_i)_{i\in[n]}, k\big)$ with $A_i = \{p \in Q_i \mid \exists \alpha' \sim \alpha\colon \alpha' \in L^\omega(\mathcal{A}_i^{p\to})\}$ for all $i \in [n]$. Then A_i is the set of states of the Büchi-automaton $\mathcal{A}_i$ that allow to accept some word from $[\alpha]$.

Since the alphabet (Γ_k, I_k) is complete, one can decide for any tuple $\big((R_i, R_i^F)_{i\in[n]}, k\big)$ whether it is the type of some word over Γ_k [2]. This decidability can be extended to infinite words, i.e., also for a tuple $\big((A_i)_{i\in[n]}, k\big)$, one can decide whether it is the type of some ω-word over Γ_k again using the completeness of (Γ_k, I_k). Note that there are only finitely many types. Hence the set of all types of mono-alphabetic words is an alphabet that can be computed. From the set of all types and the tuple of Büchi-automata, one can construct *type automata* $\mathcal{A}_i^{\mathrm{mono}}$ and $\mathcal{A}_i^{\mathrm{multi}}$ over the set of all types with the following central properties:

- Suppose $u_1, u_2, \dots \in \Gamma^+$ are mono-alphabetic words. Then the infinite type sequence $\mathrm{tp}_+(u_1)\,\mathrm{tp}_+(u_2) \cdots$ is accepted by the Büchi-automaton $\mathcal{A}_i^{\mathrm{multi}}$ iff $u_1u_2\cdots \in [L^\omega(\mathcal{A}_i)]$ and none of the factors u_iu_{i+1} is mono-alphabetic (cf. [21, Lemma 4.3]).
- Suppose $u_1, \dots, u_\ell \in \Gamma^+$ and $\beta \in \Gamma^\infty$ are mono-alphabetic words. Then the finite type sequence $\mathrm{tp}_+(u_1)\,\mathrm{tp}_+(u_2) \cdots \mathrm{tp}_+(u_\ell)\,\mathrm{tp}_\omega(\beta)$ is accepted by the nfa $\mathcal{A}_i^{\mathrm{mono}}$ iff $u_1u_2\cdots u_\ell\beta \in [L^\omega(\mathcal{A}_i)]$ and none of the factors u_iu_{i+1} nor $u_\ell\,\beta$ is mono-alphabetic.

Using classical constructions from automata theory, one can construct an nfa $\mathcal{B}^{\mathrm{mono}}$ and a Büchi automaton $\mathcal{B}^{\mathrm{multi}}$ over the set of all types accepting $\big\langle L^*(\mathcal{A}_1^{\mathrm{mono}}), \dots, L^*(\mathcal{A}_1^{\mathrm{mono}})\big\rangle^\varphi$ and $\big\langle L^\omega(\mathcal{A}_1^{\mathrm{multi}}), \dots, L^\omega(\mathcal{A}_1^{\mathrm{multi}})\big\rangle^\varphi$, respectively.

Consider some transition in the Büchi-automaton $\mathcal{B}^{\mathrm{multi}}$. Its label is the type t_+ of some finite mono-alphabetic word over Γ_k. Since the subalphabet (Γ_k, I_k) is complete, one can construct an nfa $\mathcal{C}_{t_+}$ such that the *closure* $[L^+(\mathcal{C}_{t_+})]$ of its language is the set of all finite words of type t_+ [2]. In the nfa $\mathcal{B}^{\mathrm{mono}}$, some of the transitions are labeled by the type t_ω of some infinite mono-alphabetic word over Γ_k. Again using the completeness of (Γ_k, I_k), one can also construct a Büchi-automaton $\mathcal{C}_{t_\omega}$ such that the *closure* $[L^\omega(\mathcal{C}_{t_\omega})]$ of its language is the set of all ω-words of type t_ω.

Now, in the type automata $\mathcal{B}^{\mathrm{mono}}$ and $\mathcal{B}^{\mathrm{multi}}$, replace every t_+-labeled transition with the nfa $\mathcal{C}_{t_+}$. Furthermore, in the nfa $\mathcal{B}^{\mathrm{mono}}$, replace every t_ω-labeled transition from p to some accepting state with the Büchi-automaton $\mathcal{B}_{t_\omega}$. Let $\mathcal{B}$ denote the disjoint union of the resulting two Büchi-automata. One then shows that the closure of the language of $\mathcal{B}$ is the given Boolean combination of the closures of the languages of $\mathcal{A}_i$. In summary, this proves

▶ **Theorem 4.1.** *From a transitive independence alphabet* (Γ, I)*, an* n*-ary Boolean function* φ*, and Büchi-automata* $\mathcal{A}_i$ *over* Γ *for* $i \in [n]$*, one can construct a Büchi-automaton* $\mathcal{B}$ *with*

$$[L^\omega(\mathcal{B})] = \big\langle [L^\omega(\mathcal{A}_1)], \dots, [L^\omega(\mathcal{A}_n)]\big\rangle^\varphi .$$

Hence, the problems $\mathrm{RatP}_\omega(\mathrm{TransA}, \mathrm{BF})$ *and* $\mathrm{EmptP}_\omega(\mathrm{TransA}, \mathrm{BF})$ *are decidable.*

Next, we consider the regularity problem $\mathrm{RegP}_\omega(\mathrm{TransA}, \mathrm{Id})$. So let, as before, (Γ, I) be the disjoint union of the complete independence alphabets (Γ_k, I_k) for $k \in [m]$ and let $\mathcal{A}$ be some Büchi-automaton over Γ. We will use types and type automata as defined above, this time for a single automaton, i.e., $n = 1$ and $\mathcal{A}_1 = \mathcal{A}$.

Two mono-alphabetic words u and v are *type-equivalent* (denoted $u \equiv_\mathcal{A} v$) if they have the same type. Then the relation $\equiv_\mathcal{A}$ is an equivalence relation on $\bigcup_{k\in[m]} \Gamma_k^\infty$ that only relates words from Γ_k^+ or from Γ_k^ω for some $k \in [m]$; such equivalence relations shall be called *multi-equivalences*. A language $L \subseteq \Gamma^\omega$ is *saturated* by the multi-equivalence $\equiv$ iff replacing maximal mono-alphabetic factors in words from L by equivalent ones results in a word from L. More precisely, iff the following hold.

- Let $u_1 u_2 \cdots \in L$ such that all factors u_i, but none of the factors $u_i u_{i+1}$ is mono-alphabetic. Furthermore, let $u_i \equiv v_i$ for all $i \geq 1$. Then $v_1 v_2 \cdots \in L$.
- Let $u_1 u_2 \cdots u_\ell \beta \in L$ such that all factors u_i and β, but none of the factors $u_i u_{i+1}$ and $u_\ell \beta$ is mono-alphabetic. Furthermore, let $u_i \equiv v_i$ and $\beta \equiv \gamma$. Then $v_1 v_2 \cdots v_\ell \gamma \in L$.

Recall that, in the definitions of the type of $x \in \Gamma_k^\infty$, we considered all words equivalent with x. Therefore, the type-equivalence $\equiv_\mathcal{A}$ saturates the closure $[L^\omega(\mathcal{A})]$ of the language of $\mathcal{A}$. Furthermore, the equivalence classes of $\equiv_\mathcal{A}$ are closures of regular languages. But, in general, these closures need not be regular. The central insight on the regularity of $[L^\omega(\mathcal{A})]$ is the following.

▶ **Proposition 4.2.** *Let $\mathcal{A}$ be a Büchi-automaton. Then $[L^\omega(\mathcal{A})]$ is regular if, and only if, $[L^\omega(\mathcal{A})]$ is saturated by some multi-equivalence $\equiv \supseteq \equiv_\mathcal{A}$ such that all equivalence classes of $\equiv$ are regular.*

Let $\equiv \supseteq \equiv_\mathcal{A}$ be a multi-equivalence and C some equivalence class of $\equiv$. Since $\equiv$ is a multi-equivalence, C is a language over some complete independence alphabet (Γ_k, I_k). Since $\equiv$ extends $\equiv_\mathcal{A}$, C is a union of equivalence classes of $\equiv_\mathcal{A}$. Note that $\equiv_\mathcal{A}$ has only finitely many equivalence classes and recall that any such equivalence class is the closure of some regular language. Hence C is the closure of the union of finitely many regular languages. If C is a set of finite words, then its regularity can be decided by [2], and also in case C is a set of infinite words, this is possible. It remains to decide whether $\equiv$ saturates the closure of $L^\omega(\mathcal{A})$. To this aim, we define the *block closure* $\mathrm{blkcl}(L)$ of a language $L \subseteq \Gamma^\omega$ as the minimal language containing L which is saturated by $\equiv$. Since the equivalence classes of $\equiv_\mathcal{A}$ are closed languages, the same holds for the equivalence classes of $\equiv$. Hence $\mathrm{blkcl}([L]) = \mathrm{blkcl}(L) = [\mathrm{blkcl}(L)]$. Since the equivalence classes of $\equiv$ are regular, one can construct a Büchi-automaton $\mathcal{B}$ that accepts $\mathrm{blkcl}(L^\omega(\mathcal{A}))$. Hence $[L^\omega(\mathcal{A})]$ is saturated by $\equiv$, i.e., $\mathrm{blkcl}([L^\omega(\mathcal{A})]) \subseteq [L^\omega(\mathcal{A})]$ if, and only if, $[L^\omega(\mathcal{B})] \subseteq [L^\omega(\mathcal{A})]$ – which is decidable using Theorem 4.1.

Hence, for any multi-equivalence $\equiv$ extending the type equivalence $\equiv_\mathcal{A}$, one can decide whether it witnesses the regularity of $[L^\omega(\mathcal{A})]$ in the sense of Prop. 4.2. Since there are only finitely many such equivalence relations and since this finite set can be computed, we obtain that the regularity of $[L^\omega(\mathcal{A})]$ is decidable. Now Theorem 4.1 implies the decidability of $\mathrm{RegP}_\omega(\mathrm{TransA}, \mathrm{BF})$: to handle an arbitrary function φ, first compute a single automaton $\mathcal{B}$ as in Theorem 4.1 and then check the regularity of the closure $[L^\omega(\mathcal{B})]$ of its language.

Hence we obtain the following extension of Theorem 3.4(2') and therefore of Sakarovich's result [28] from finite words to ω-words.

▶ **Theorem 4.3.** *The regularity problem $\mathrm{RegP}_\omega(\mathrm{TransA}, \mathrm{BF})$ is decidable.*

5 Diagonal independence alphabets

In this section, we consider the ω-versions of the emptiness, the rationality, and the regularity problem for diagonal independence alphabets. For the emptiness problem, we will distinguish the case of monotone and non-monotone Boolean functions since the problem becomes decidable in the former case and undecidable in the latter. For the other two problems, the monotonicity of the Boolean function does not play any role.

Emptiness for monotone Boolean functions

The aim of this section is to show that the emptiness problem $\mathrm{EmptP}_\omega(\mathrm{DiagA}, \mathrm{monBF})$ for diagonal independence alphabets and monotone Boolean functions is decidable. Since $\mathrm{EmptP}_*(\mathrm{DiagA}, \mathrm{BF})$ is undecidable (cf. Theorem 3.2(2)), the crucial point in the decidability proof of $\mathrm{EmptP}_\omega(\mathrm{DiagA}, \mathrm{monBF})$ is the monotonicity of the Boolean functions considered.

A first simple observation is that it suffices to consider connected diagonal independence alphabets; we denote the set of all such independence alphabets with ConDiagA. These independence alphabets can be considered as *order trees*, i.e., a structure $(\Gamma, \preceq)$ where $\preceq$ is a partial order on the finite set Γ with largest element such that, for all $a, b, c \in \Gamma$, $c \preceq a, b$ implies $a \preceq b$ or $b \preceq a$. An independence alphabet (Γ, I) is connected and diagonal if, and only if, there exists an order tree $(\Gamma, \preceq)$ with $(a, b) \in I \iff a \prec b$ or $b \prec a$, i.e., $I = \succ \cup \prec$ [29, 30].

In the following, we will always assume that $(\Gamma, \preceq)$ is an order tree with $I = \succ \cup \prec$.

For $a \in \Gamma$, let $\downarrow\! a = \{b \in \Gamma \mid b \preceq a\}$ and $\Downarrow\! a = \{b \in \Gamma \mid b \prec a\}$ be the sets of elements of Γ (properly) below a; $\updownarrow\! a \subseteq \Gamma$ is the set of letters comparable with a, i.e., a together with all letters $b \in \Gamma$ with $(a, b) \in I$.

We next define relations that are central to our arguments, these definitions are adaptations of similar notions from [1] to Büchi-automata. Let $\overline{\mathcal{A}} = (\mathcal{A}_i)_{i \in [n]}$ be a tuple of nfas $\mathcal{A}_i = (Q_i, \Gamma, I_i, T_i, F_i)$, $\overline{p} = (p_i)_{i \in [n]}$ and $\overline{q} = (q_i)_{i \in [n]}$ tuples of states from $\prod_{i \in [n]} Q_i$, and $a \in \Gamma$. Then $R(\overline{p}, \overline{q}, a)$ denotes the set of tuples of words that lead from p_i to q_i visiting some accepting state and use only letters independent from or equal to a:

$$R(\overline{p}, \overline{q}, a) = \left\{(u_1, \dots, u_n) \;\middle|\; u_i \in \updownarrow\! a^*\,,\; p_i \overset{u_i}{\Longrightarrow}_{\mathcal{A}_i} q_i \text{ for all } i \in [n]\right\}$$

Collecting all tuples of ω-words over $\updownarrow\! a$ accepted by $\mathcal{A}_i$ from p_i, we get the analogous relation

$$R^\omega(\overline{p}, a) = \left\{(\alpha_1, \dots, \alpha_n) \;\middle|\; \alpha_i \in \updownarrow\! a^\omega \cap L^\omega(\mathcal{A}_i^{p_i \to}) \text{ for all } i \in [n]\right\}.$$

We will be interested in the restrictions of these relations to tuples of words whose projections to $\Downarrow\! a$ and $\downarrow\! a$ are mutually equivalent:

$$\begin{aligned}
R_{\Downarrow}(\overline{p}, \overline{q}, a) &= \left\{(u_i)_{i \in [n]} \in R(\overline{p}, \overline{q}, a) \;\middle|\; \mathrm{proj}_{\Downarrow a}(u_1) \sim \mathrm{proj}_{\Downarrow a}(u_i) \text{ for all } i \in [n]\right\}\\
R^\omega_{\downarrow}(\overline{p}, a) &= \left\{(\alpha_i)_{i \in [n]} \in R^\omega(\overline{p}, a) \;\middle|\; \mathrm{proj}_{\downarrow a}(\alpha_1) \sim \mathrm{proj}_{\downarrow a}(\alpha_i) \text{ for all } i \in [n]\right\}\\
R^\omega_{\Downarrow}(\overline{p}, a) &= \left\{(\alpha_i)_{i \in [n]} \in R^\omega(\overline{p}, a) \;\middle|\; \mathrm{proj}_{\Downarrow a}(\alpha_1) \sim \mathrm{proj}_{\Downarrow a}(\alpha_i) \text{ for all } i \in [n]\right\}
\end{aligned}$$

Suppose a is the root of $(\Gamma, \preceq)$ such that $\mathrm{proj}_{\downarrow a}(\alpha) = \alpha$ for all $\alpha \in \Gamma^\omega$. Furthermore, suppose $\varphi(x_1, \dots, x_n) = \min\{x_1, \dots, x_n\}$, and let p_i be the (only) initial state of $\mathcal{A}_i$ for all $i \in [n]$. Then $\left\langle [L^\omega(\mathcal{A}_1)], \dots, [L^\omega(\mathcal{A}_n)]\right\rangle^\varphi$ equals $R^\omega_{\downarrow}(\overline{p}, a)$, so we have to decide the emptiness of this relation. If it was rational, this would easily be possible. Unfortunately, this is not the case in general. For finite words, Aalbersberg and Hoogeboom approximate the relation $R_{\Downarrow}(\overline{p}, \overline{q}, a)$ by its Parikh image. In order to also handle ω-words, we generalize this by the notion of a "language version" that is introduced next.

For an alphabet Σ (e.g., $\Sigma = \Gamma \times [n]$), the Parikh mapping $\Psi\colon \Sigma^\infty \to (\mathbb{N} \cup \{\infty\})^\Sigma$ is defined by $\Psi(\alpha) = (|\alpha|_a)_{a \in \Sigma}$ for all $\alpha \in \Sigma^\infty$. We extend the Parikh mapping to tuples of words: the mapping $\Psi_n\colon (\Gamma^\infty)^n \to (\mathbb{N} \cup \{\infty\})^{\Gamma \times [n]}$ is defined by $\Psi_n(x_1, \dots, x_n)_{(a,i)} = |x_i|_a$, i.e., $\Psi_n(x_1, \dots, x_n)(a, i)$ is the number of occurrences of the letter a in the word number i. The language $L \subseteq (\Gamma \times [n])^\infty$ is a *language version* of the relation $R \subseteq (\Gamma^\infty)^n$ if $\Psi_n(R) = \Psi(L)$.

The following crucial result will allow us to decide the problem $\mathrm{EmptP}_\omega(\mathrm{DiagA}, \mathrm{monBF})$.

▶ **Lemma 5.1.** *One can construct a Büchi-automaton $\mathcal{B}$ that accepts a language version of the relation $R = \left\{(\alpha_i)_{i \in [n]} \mid \alpha_i \in L^\omega(\mathcal{A}_i) \text{ and } \alpha_1 \sim \alpha_i \text{ for all } i \in [n]\right\}$.*

Proof idea. In addition to the above relations $R_{\Downarrow}(\overline{p}, \overline{q}, a)$ etc., we will also make use of the following relation that collects all tuples of ω-words that agree in their number of occurrences of a:

$$\mathrm{Eq}^\omega_a = \left\{(\alpha_i)_{i \in [n]} \in (\Gamma^\omega)^n \colon |\alpha_1|_a = |\alpha_i|_a \text{ for all } i \in [n]\right\}$$

Note that this relation does not depend on the automaton $\mathcal{B}$. By induction on the size of $\downarrow a$, one shows $R^\omega_\downarrow(\overline{p}, a) = R^\omega_\Downarrow(\overline{p}, a) \cap \mathrm{Eq}^\omega_a$ and that the relation $R^\omega_\Downarrow(\overline{p}, a)$ can be constructed from the relations $R_\Downarrow(\overline{q}, \overline{r}, a)$ and $R^\omega_\downarrow(\overline{q}, b)$ for $b \prec a$ using union, concatenation, and ω-iteration.

From [1], one gets regular language versions of the relations $R_\Downarrow(\overline{p}, \overline{q}, a)$. Note that the union of language versions of two relations is a language version of the union (similarly for concatenation and iteration). While this does not hold for all intersections of relations, we can nevertheless construct a language version for the intersection with the very special relation Eq^ω_a and its language version of all ω-words over $\Gamma \times [n]$ whose projection to $\{a\} \times [n]$ belongs to $\big((a,1)(a,2)\dots(a,n)\big)^\infty$. To handle this intersection, the idea is to consider the complete independence relation on $\Gamma \times [n]$ and then to refer to Theorem 4.1. ◀

Since the emptiness of the regular language version of R is decidable, one can infer the main decidability result for diagonal independence alphabets; it is an extension of Lemma 3.1(3) (which follows easily from [1]) to Büchi-automata.

▶ **Theorem 5.2.** *The emptiness problem* $\mathrm{EmptP}_\omega(\mathrm{DiagA}, \mathrm{monBF})$ *is decidable.*

Emptiness for general Boolean functions

Next, we study non-monotone Boolean functions φ. The Π^0_1-hardness of $\mathrm{EmptP}_*(\mathrm{DiagA}, \varphi)$ (Theorem 3.2(2)) transfers to $\mathrm{EmptP}_\omega(\mathrm{DiagA}, \varphi)$. To show containment in Π^0_1, one generalizes the proof of [21, Cor. 5.5] that yields the result for the Boolean function $\varphi(x_1, x_2) = x_1 \wedge \neg x_2$.

The crucial insight is that, with $H = \big\langle [L^\omega(\mathcal{A}_1)], \dots, [L^\omega(\mathcal{A}_n)] \big\rangle^\varphi$ and the independence alphabet diagonal, $H = \emptyset$ or H contains some ultimately periodic ω-word uv^ω (this is not necessarily the case if the independence alphabet is not diagonal).

Since it is decidable whether an ultimately periodic word is in $[L^\omega(\mathcal{A}_i)]$ [21, Obs. 5.1], we can just search for a witness for non-emptiness. It follows that the non-emptiness problem is semi-decidable.

▶ **Corollary 5.3.** *The emptiness problem* $\mathrm{EmptP}_\omega(\mathrm{DiagA}, \mathrm{BF})$ *is in* Π^0_1.

Rationality and regularity problem

By Theorem 3.4, the problems $\mathrm{RatP}_*((\Gamma, I), \neg)$ and $\mathrm{RegP}_*((\Gamma, I), \neg)$ are Σ^0_2-hard for any non-transitive independence alphabet (Γ, I), these lower bounds transfer to the ω-versions of these problems. We can also prove matching upper bounds for the set of all diagonal independence alphabets and all Boolean functions. These proofs rely on the emptiness problem and are almost identical to the corresponding proof for finite words, cf. Lemma 3.3(2).

▶ **Theorem 5.4.** *The rationality problem* $\mathrm{RatP}_\omega(\mathrm{DiagA}, \mathrm{BF})$ *and the regularity problem* $\mathrm{RegP}_\omega(\mathrm{DiagA}, \mathrm{BF})$ *are in* Σ^0_2.

6 General independence alphabets

The problems $\mathrm{EmptP}_\omega((\Gamma, I), \varphi)$ for $\varphi \in \{\wedge, \neg\}$ and $\mathrm{RegP}_\omega((\Gamma, I), \mathrm{Id})$ are Π^1_1-hard for any non-diagonal independence alphabet (Γ, I) [21, Thm. 6.4]. In view of Lemma 2.4, it follows that $\mathrm{EmptP}_\omega(\mathbb{A}, \Phi)$ (and $\mathrm{RegP}_\omega(\mathbb{A}, \Phi)$) is Π^1_1-hard if $\mathbb{A} \not\subseteq \mathrm{DiagA}$ and $\Phi \not\subseteq \mathrm{MAX}$ ($\Phi \not\subseteq \mathrm{CONST}$, resp.)

Here, we indicate why also the rationality problem $\mathrm{RatP}_\omega(\mathbb{A}, \Phi)$ is Π^1_1-hard in case $\mathbb{A}$ contains some non-diagonal independence alphabet and Φ some non-max function. We only sketch the arguments for the Π^1_1-hardness of the rationality problem $\mathrm{RatP}_\omega((\Sigma, I), \wedge)$ (and later for the rationality problem $\mathrm{RatP}_\omega((\Sigma, I), \neg))$ for one particular independence alphabet: $\mathrm{Op} = \{\mathrm{inc}_k, \mathrm{dec}_k, \mathrm{zero}_k \mid k \in [2]\}$ is the set of the "counter operations" increment, decrement, and zero-test for the counters 1 and 2, $\Sigma = \mathrm{Op} \cup \{b, c, d\}$, and $\mathrm{Op} \times \{b, d\} \subseteq I$, $\mathrm{Op} \times \{c\} \subseteq D$, $(b, c), (c, d) \in I$, and $(b, d) \in D$ (note that the independence alphabet (Σ, I) is not diagonal since, e.g., inc_1, b, c, d induce a C_4).

The proof modifies the proofs from [21] (that heavily rely on ideas from [1]); hence we recall relevant parts from [21] and indicate the additions that are relevant for the rationality problem.

One starts with non-deterministic two-counter machines with Büchi-acceptance, i.e., Büchi-automata over the alphabet of counter operations Op. To define the executable or valid words, let $\alpha = a_0 a_1 \cdots \in \mathrm{Op}^\omega$ with $a_i \in \mathrm{Op}$ for all $i \geq 0$. For $n \geq 0$ and $k \in [2]$, let $\mathrm{value}_k(\alpha, n) = |a_0 a_1 \cdots a_{n-1}|_{\mathrm{inc}_k} - |a_0 a_1 \cdots a_{n-1}|_{\mathrm{dec}_k}$ denote the value of the counter k after executing the first n operations of α. The ω-word α is *valid* if, for all $n \geq 0$ and $k \in [2]$, $\mathrm{value}_k(\alpha, n) = 0$ whenever $a_n = \mathrm{zero}_k$ and $\mathrm{value}_k(\alpha, n) > 0$ whenever $a_n = \mathrm{dec}_k$. Let $\mathrm{Val} \subseteq \mathrm{Op}^\omega$ denote the set of all valid ω-words.

Based on [17], one then obtains that the set of Büchi-automata $\mathcal{M}$ over Op with $L^\omega(\mathcal{M}) \cap \mathrm{Val} = \emptyset$ is Π^1_1-complete [21, Thm. 6.1]. To reduce this problem to the rationality problem $\mathrm{RatP}_\omega((\Sigma, I), \wedge)$, let $\mathcal{M}$ be some Büchi-automaton over Op. We first let $H = L^\omega(\mathcal{M}) \sqcup \{b, c, d\}^\infty$ denote the regular set of words over Σ whose projection to Op is accepted by $\mathcal{M}$.

One then defines two regular ω-languages K and L over Σ (that do not depend on $\mathcal{M}$) as follows: $K_0 = \mathrm{Op}\, d\, (cb)^+$, $K = b\, K_0^\omega$,

$$
\begin{aligned}
L_0 = {} & \mathrm{inc}_1\, (b\, c^5\, c^5)^+\, d \cup \mathrm{inc}_2\, (b\, c^5\, c^5\, c^5)^+\, d \\
& \cup \mathrm{dec}_1\, (b\, b\, c^5)^+\, d \cup \mathrm{dec}_2\, (b\, b\, b\, c^5)^+\, d \\
& \cup \mathrm{zero}_1\, (b\, c^5\, b\, c^5)^*\, b\, c^5\, d \cup \mathrm{zero}_2\, (b\, c^5\, b\, c^5\, b\, c^5)^*\, \{b\, c^5\, b\, c^5, b\, c^5\}\, d\,,
\end{aligned}
$$

and $L = L_0^\omega$. Note that, in any word from K_0, the number of occurrences of c and of b coincide. Differently, in words from L_0, these numbers differ depending on the letter from Op the word starts with. In [21], it is demonstrated that the projection of the language $[K] \cap [L]$ to $\mathrm{Op} \cup \{c\}$ consists of all ω-words $a_0\, c^{m_1}\, a_1\, c^{m_2} \cdots$ (with $a_i \in \mathrm{Op}$) such that

$$
m_{i+1} = 2^{\mathrm{value}_1(a_0 a_1 \cdots a_i)} \cdot 3^{\mathrm{value}_2(a_0 a_1 \cdots a_i)} \cdot 5^i
$$

as well as $2 \not| m_{i+1}$ and $3 \not| m_{i+1}$ in case $a_i = \mathrm{zero}_1$ and $a_i = \mathrm{zero}_2$, respectively. In other words, the length of the block $c^{m_{i+1}}$ encodes the counter values after executing $a_0 a_1 \cdots a_i$ (and verifies that they are 0 if a_i requires this). Dropping also the occurrences of the letter c from this projection, we therefore obtain precisely the valid words: $\mathrm{Val} = \mathrm{proj}_{\mathrm{Op}}([K] \cap [L])$ [21, Prop. 6.2].

Note that we always have $m_i < m_{i+1}$ since we did not only encode the counter values (as exponents of 2 and 3, resp.), but also the step counter as exponent of 5. If follows that none of the words from $\mathrm{proj}_{\mathrm{Op} \cup \{c\}}([K] \cap [L])$ is ultimately periodic implying that $[K] \cap [L]$ contains no ultimately periodic word [21, Prop. 6.2].

We can now relate the emptiness of $L^\omega(\mathcal{M}) \cap \mathrm{Val}$ to the existence of a regular language whose closure is the intersection of the closures of $H \cap K$ and L.

▶ **Lemma 6.1.** *The following are equivalent:* **(i)** $L^\omega(\mathcal{M}) \cap \mathrm{Val} = \emptyset$. **(ii)** $[H \cap K] \cap [L] = \emptyset$. **(iii)** $[M] = [H \cap K] \cap [L]$ *for some regular language* M.

Note that, differently from [21], statement (iii) does not require $[H \cap K] \cap [L]$ to be regular, but only to be the closure of some regular language.

Proof. Since the language H is closed, we have $[H \cap K] \cap [L] = H \cap [K] \cap [L]$ and therefore $\mathrm{proj}_{\mathrm{Op}}([H \cap K] \cap [L]) = L^\omega(\mathcal{M}) \cap \mathrm{Val}$ which proves the equivalence of (i) and (ii) (this was already shown in [21]).

The implication (ii)⇒(iii) is trivial, just set $M = \emptyset$.

The final implication (iii)⇒(ii) is shown by contradiction. So suppose (iii) holds, but $[H \cap K] \cap [L] \neq \emptyset$. In other words, there is a non-empty regular language $M \subseteq \Gamma^\omega$ with $[H \cap K] \cap [L] = [M]$. Since M is regular, there are nonempty words $u, v \in \Gamma^+$ with $uv^\omega \in M$. But then $uv^\omega \in M \subseteq [M] = [H \cap K] \cap [L] \subseteq [K] \cap [L]$ which contradicts the above observation that this latter set does not contain any ultimately periodic word. ◀

Finally note that, from the Büchi-automaton $\mathcal{M}$, one can construct Büchi-automata accepting $H \cap K$ and L, resp. Hence the equivalence of (i) and (iii) above provides a reduction of the Π^1_1-complete question "$L^\omega(\mathcal{M}) \cap \mathrm{Val} = \emptyset$?" to the rationality problem $\mathrm{RatP}_\omega((\Sigma, I), \wedge)$.

We next sketch the hardness proof for the Boolean function $\neg$ in place of $\wedge$. Here, one constructs (from $\mathcal{M}$) a Büchi-automaton $\mathcal{A}$ such that $\Sigma^\omega \setminus [L^\omega(\mathcal{A})] = [H \cap K] \cap [L]$. This construction reduces the Π^1_1-complete question "$L^\omega(\mathcal{M}) \cap \mathrm{Val} = \emptyset$?" to the rationality problem $\mathrm{RatP}_\omega((\Sigma, I), \neg)$.

Using some coding tricks, one can reduce these two problems $\mathrm{RatP}_\omega((\Sigma, I), \varphi)$ for $\varphi \in \{\wedge, \neg\}$ to the problems $\mathrm{RatP}_\omega((\Gamma, I), \varphi)$ provided (Γ, I) contains some C_4.

The above arguments hold true as well if, in the independence alphabet (Σ, I), we set $\mathrm{Op} \times \{d\} \subseteq D$ (instead of $\mathrm{Op} \times \{d\} \subseteq I$). It follows that these problems are also Π^1_1-hard for any P_4-free independence alphabet. Finally, referring to Lemma 2.4, one obtains the following lower bounds. The upper bounds are rather straightforward consequences of the definition of these problems.

▶ **Theorem 6.2.**

(1) *For all* $\mathrm{GenP} \in \{\mathrm{EmptP}, \mathrm{RatP}, \mathrm{RegP}\}$*, the problem* $\mathrm{GenP}_\omega(\mathrm{IndA}, \mathrm{BF})$ *is in* Π^1_2.
The problem $\mathrm{EmptP}_\omega(\mathrm{IndA}, \mathrm{monBF})$ *is in* Π^1_1.

(2) *Let* $\mathbb{A} \subseteq \mathrm{IndA}$ *with* $\mathbb{A} \not\subseteq \mathrm{DiagA}$ *and* $\Phi \subseteq \mathrm{BF}$ *be decidable sets.*
If $\Phi \not\subseteq \mathrm{MAX}$*, then* $\mathrm{RatP}_\omega(\mathbb{A}, \Phi)$ *and* $\mathrm{EmptP}_\omega(\mathbb{A}, \Phi)$ *are* Π^1_1*-hard.*
If $\Phi \not\subseteq \mathrm{CONST}$*, then* $\mathrm{RegP}_\omega(\mathbb{A}, \Phi)$ *is* Π^1_1*-hard.*

7 Summary and open problems

A discussion of the results of this paper (cf. Table 1) can be found at the end of the introduction. In particular, we solved two of the open problems from [21]: the regularity problem for transitive independence alphabets and the emptiness problem for diagonal independence alphabets and monotone Boolean functions are decidable and not just semi-decidable.

Let (Γ_1, I_1) and (Γ_2, I_2) be two non-diagonal independence alphabets and φ_1, φ_2 be two monotone, but non-max functions. Then the emptiness problems $\mathrm{EmptP}_\omega((\Gamma_1, I_1), \varphi_1)$ and $\mathrm{EmptP}_\omega((\Gamma_2, I_2), \varphi_2)$ both are Π^1_1-complete and therefore inter-reducible. If φ_1 and φ_2 are non-monotone, containment in Π^1_1 is not known. We do not even know whether the two problems are inter-reducible or whether these problems define different (possibly many) degrees of unsolvability.

References

1 I.J. Aalbersberg and H.J. Hoogeboom. Characterizations of the decidability of some problems for regular trace languages. *Math. Syst. Theory*, 22(1):1–19, 1989. doi:10.1007/BF02088289.

2 I.J. Aalbersberg and E. Welzl. Trace languages defined by regular string languages. *RAIRO Theor. Informatics Appl.*, 20(2):103–119, 1986. doi:10.1051/ita/1986200201031.

3 B. Adsul, P. Gastin, S. Kulkarni, and P. Weil. An expressively complete local past propositional dynamic logic over Mazurkiewicz traces and its applications. In *LICS'24*, pages 2:1–2:13. ACM, 2024. doi:10.1145/3661814.3662110.

4 J. Berstel. *Transductions and context-free languages*. Teubner, Stuttgart, 1979.

5 D. Brand and P. Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983. doi:10.1145/322374.322380.

6 S. Breuers, Ch. Löding, and J. Olschewski. Improved Ramsey-based Büchi complementation. In *FoSSaCS'12*, Lecture Notes in Comp. Science vol. 7213, pages 150–164. Springer, 2012. doi:10.1007/978-3-642-28729-9_10.

7 V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, 1995. doi:10.1142/2563.

8 P. Gastin. Infinite traces. In *Semantics of Systems of Concurrent Processes*, Lecture Notes in Comp. Science vol. 469, pages 277–308. Springer, 1990. doi:10.1007/3-540-53479-2_12.

9 P. Gastin. Recognizable and rational languages of finite and infinite traces. In *STACS'91*, Lecture Notes in Comp. Science vol. 480, pages 89–104. Springer, 1991. doi:10.1007/BFb0020790.

10 P. Gastin and D. Kuske. Uniform satisfiability in PSPACE for local temporal logics over Mazurkiewicz traces. *Fundam. Informaticae*, 80(1-3):169–197, 2007. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi80-1-3-10.

11 P. Gastin, B. Lerman, and M. Zeitoun. Distributed games with causal memory are decidable for series-parallel systems. In *FSTTCS'04*, Lecture Notes in Comp. Science vol. 3328, pages 275–286. Springer, 2004. doi:10.1007/978-3-540-30538-5_23.

12 B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Inf. Comput.*, 204(6):920–956, 2006. doi:10.1016/j.ic.2006.01.005.

13 B. Genest, D. Kuske, and A. Muscholl. On communicating automata with bounded channels. *Fundam. Informaticae*, 80(1-3):147–167, 2007. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi80-1-3-09.

14 Hugo Gimbert. On the control of asynchronous automata. In *FSTTCS'17*, volume 93 of *LIPIcs*, pages 30:1–30:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.30.

15 Hugo Gimbert. Distributed asynchronous games with causal memory are undecidable. *Log. Methods Comput. Sci.*, 18(3), 2022. doi:10.46298/lmcs-18(3:30)2022.

16 F. Gire and M. Nivat. Relations rationelles infinitaires. *Calcolo*, 21:91–125, 1984.

17 D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM*, 33(1):224–248, 1986. doi:10.1145/4904.4993.

18 M. Hutagalung, N. Hundeshagen, D. Kuske, M. Lange, and É. Lozes. Multi-buffer simulations: decidability and complexity. *Inf. Comput.*, 262(2):280–310, 2018. doi:10.1016/j.ic.2018.09.008.

19 Ch. Köcher and D. Kuske. Backwards-reachability for cooperating multi-pushdown systems. *J. Comput. Syst. Sci.*, 148:103601, 2025. doi:10.1016/j.jcss.2024.103601.

20 Ch. Köcher and D. Kuske. Reachability in trace-pushdown systems. *Theor. Comput. Sci.*, 1077:115971, 2026. doi:10.1016/j.tcs.2026.115971.

21 D. Kuske. Disjointness, inclusion, and regularity of ω-rational trace languages – extended abstract –. In *FCT'25*, Lecture Notes in Comp. Science vol. 16106, pages 322–335. Springer, 2025. doi:10.1007/978-3-032-04700-7_24.

22 D. Kuske. The theory of reachability in trace-pushdown systems. *CoRR*, abs/2507.15733, 2025. conditionally accepted by Fundamenta Informaticae. doi:10.48550/arXiv.2507.15733.

23 D. Kuske and A. Muscholl. Communicating automata. In J.-E. Pin, editor, *Handbook of Automata Theory*, volume 2, pages 1147–1188. European Mathematical Society Publishing House, Zürich, Switzerland, 2021. doi:10.4171/Automata-2/9.

24 L. P. Lisovik. The identity problem for regular events over the direct product of free and cyclic semigroups. *Dok. Akad. Nau Ukraiinskoj RSR sers A*, 6:410–413, 1979. in Ukrainian.

25 A. Mazurkiewicz. Concurrent program schemes and their interpretation. Technical report, DAIMI Report PB-78, Aarhus University, 1977.

26 A. Muscholl and H. Petersen. A note on the commutative closure of star-free languages. *Inf. Process. Lett.*, 57(2):71–74, 1996. doi:10.1016/0020-0190(95)00187-5.

27 P. Odifreddi. *Classical recursion theory. The theory of functions and sets of natural numbers*, volume 125 of *Stud. Logic Found. Math.* Amsterdam etc. North-Holland, 1989.

28 J. Sakarovitch. The "last" decision problem for rational trace languages. In *LATIN'92*, Lecture Notes in Comp. Science vol. 583, pages 460–473. Springer, 1992. doi:10.1007/BFb0023848.

29 E.S. Wolk. The comparability graph of a tree. *Proceedings of the AMS*, 13:789–795, 1962.

30 E.S. Wolk. A note on "The comparability graph of a tree". *Proceedings of the AMS*, 16(1):17–20, 1965.