

Unfolding of finite concurrent automata

Alexandre Mansard

LIM - University of La Réunion
France

alexandre.mansard@univ-reunion.fr

We consider recognizable trace rewriting systems with level-regular contexts (RTL). A trace language is level-regular if the set of Foata normal forms of its elements is regular. We prove that the rewriting graph of a RTL is word-automatic. Thus its first-order theory is decidable. Then, we prove that the concurrent unfolding of a finite concurrent automaton with the reachability relation is a RTL graph. It follows that the first-order theory with the reachability predicate (FO[Reach] theory) of such an unfolding is decidable. It is known that this property holds also for the ground term rewriting graphs. We provide examples of finite concurrent automata of which the concurrent unfoldings fail to be ground term rewriting graphs. The infinite grid tree (for each vertex of an infinite grid, there is an edge from this vertex to the origin of a copy of the infinite grid) is such an unfolding. We prove that the infinite grid tree is not a ground term rewriting graph. We have thus obtained a new class of graphs for which a decidable FO[Reach] theory.

1 Introduction

A challenging problem in automatic verification consists in determining (or in extending) classes of infinite graphs having a decidable theory in a given logic. A first technique consists in considering some judicious graph transformations, as for example unfolding (that preserves decidability of monadic second-order logic) or logical interpretations. The pushdown hierarchy [3] is a hierarchy of decidable graphs of monadic second-order theory. Starting from finite graphs, each level consists of the monadic interpretations of the unfoldings of lower levels. The tree-automatic hierarchy [5] is a hierarchy of graphs of decidable first-order (FO) theory: each level consists of finite set interpretations of the corresponding level of the pushdown hierarchy. A second technique is to consider graphs whose vertex set and relations are recognizable by automata whose recognized languages form a Boolean algebra. For instance, it is the case of word-automatic graphs or more generally tree-automatic graphs, *i.e.* graphs whose vertex set can be encoded by a regular tree language and each relation recognized by a synchronized tree transducer. It turns out the first level of the tree-automatic hierarchy consists of tree-automatic graphs. Lastly, rewriting systems also allow to define interesting graph classes. Graphs at the first level of the pushdown hierarchy are the suffix rewriting graphs of recognizable word rewriting systems [2]. Ground term rewriting graphs (GTR graphs) with the reachability relation are tree-automatic and thus the first-order theory with the reachability predicate (FO[Reach]) of a GTR graph is decidable [6].

Since its monadic second-order theory is not decidable, the infinite grid does not belong to the pushdown hierarchy and is therefore not the unfolding of a finite graph. Nevertheless, as a GTR graph, the infinite grid has a decidable FO[Reach] theory. In fact, even the theory of the infinite grid in first-order logic extended by the operator of transitive closure for first-order definable relations remains decidable [15]. But consider now the infinite grid tree: from each vertex of an infinite grid, there is an edge (labelled by a new symbol) to the origin of a copy of the infinite grid. We will prove that this simple graph (it is just the configuration graph of a system with 2 counters that we can independently increment and

simultaneously reset) has the FO[Reach] theory decidable but is not a GTR graph. In fact, we are interested in considering, more generally, a class of graphs that model concurrent system computations. For such a system, sequential and parallel computations are possible. To that end, we will consider Mazurkiewicz traces: if the dependency is total, then a trace reduces to a string that describes sequential computation while independence between some letters bring the possibility to describe parallel computation.

For a recognizable trace rewriting system, that is a finite set of rules of the form $\mathcal{U} \cdot (\mathcal{V} \xrightarrow{\lambda} \mathcal{W})$ where $\mathcal{U}, \mathcal{V}, \mathcal{W}$ are recognizable trace languages, λ a label, consider then its rewriting graph: the set of edges of the form $ts \xrightarrow{\lambda} ts'$ such that there exists a rewriting rule $\mathcal{U} \cdot (\mathcal{V} \xrightarrow{\lambda} \mathcal{W})$ with $t \in \mathcal{U}, s \in \mathcal{V}, s' \in \mathcal{W}$. If all letters are dependent, then such a graph is at the first level of the pushdown hierarchy since it is the suffix rewriting graph of a recognizable word rewriting system, and if no distinct letters are dependent, then it is the configuration graph of a vector addition system. In any case, we will prove in Section 3 that such a graph is word-automatic, even with level-regular contexts: a trace language is level-regular if the set of Foata normal forms of its elements is regular. Since the set of Foata normal forms is regular, every recognizable trace language is level-regular. But, for example, if a and b are two independent letters, the trace language $[(ab)^*]$ is level-regular but not recognizable. The FO theory of the rewriting graph of a recognizable trace rewriting system with level-regular contexts (RTL graph) is thus decidable. We also prove that, in general, its FO[Reach] theory is not decidable. Otherwise, we could decide the halting problem for 2-counters Minsky machine. In Section 4, we prove that the concurrent unfolding of a finite concurrent automaton has the FO[Reach] theory decidable, by showing that such a graph with the reachability relation is a RTL graph. This extends a theorem of Madhusudan [11] on decidability of FO theory of regular trace event structures [14]. We will observe that the infinite grid and the infinite grid tree are the concurrent unfoldings of finite concurrent automata. In Section 5, we define the tree of a graph and we prove that if it is a GTR graph, then it is finitely decomposable by size. The latter implies it is at the first level of the pushdown hierarchy. We deduce that the infinite grid tree is not a GTR graph.

2 Preliminaries

Before presenting the rewriting graphs of recognizable trace rewriting systems, we recall some basic definitions about graphs, logics, automata and traces.

Let Σ be a finite alphabet and Σ^* be the free monoid of words over Σ .

2.1 Graphs

A Σ -graph G is a subset of $V \times \Sigma \times V$ where V is a set. An element $(p, a, q) \in G$ is an edge labelled by a from source p to target q . The notation $p \xrightarrow[a]{G} q$ (or $p \xrightarrow{a} q$ when G is understood) means $(p, a, q) \in G$.

The vertex set of G is $V_G = \{p \in V \mid \exists q (p \xrightarrow[a]{G} q \vee q \xrightarrow[a]{G} p)\}$.

The graph G is *deterministic* if for every $a \in \Sigma$, if $(p \xrightarrow[a]{G} q$ and $p \xrightarrow[a]{G} q')$ then $q = q'$.

A *path* in G between vertices p and q , labelled by a word $u = a_1 \dots a_k$ is a finite sequence of the form $p \xrightarrow[a_1]{G} p_1, \dots, p_{k-1} \xrightarrow[a_k]{G} q$. We denote by $p \xrightarrow[a_1 \dots a_k]{G} q$ the existence of such a path. A *loop* in G is a path of the form $p \xrightarrow[a]{G} p$, where $a \in \Sigma$. If G is finite, then for $p, q \in V_G$, the Σ -word language $L_{p,q} := \{u \in \Sigma^* \mid p \xrightarrow[u]{G} q\}$ is regular. We write $p \xrightarrow[*]{G} q$ if there exists a word $u \in \Sigma^*$ such that $p \xrightarrow[u]{G} q$. Denote by G_* the $\Sigma \cup \{*\}$ -graph

defined by $G_* = G \cup \{(p, *, q) \mid p \xrightarrow[G]{*} q\}$. The graph G_* is obtained from G by adding the reachability relation.

An isomorphism f from (G, P) onto (H, Q) , where G and H are Σ -graphs and P and Q are subsets of V_G and V_H respectively, is a bijection from V_G to V_H such that

$$f(P) = Q \text{ and } (p \xrightarrow[G]{a} q \iff f(p) \xrightarrow[H]{a} f(q)).$$

2.2 Logics

A Σ -graph G is a relational structure over the binary signature Σ . The *first-order* (FO) theory of G is defined as usual (see [7]). The FO theory of G_* will be referred to as the FO[Reach] theory of G .

2.3 Automata

A Σ -automaton is a triple $\mathcal{A} = (G, i, F)$ where G is a Σ -graph, $i \in V_G$ is an initial state and $F \subseteq V_G$ is the set of final states. The Σ -word language recognized by \mathcal{A} is $L(\mathcal{A}) = \{u \in \Sigma^* \mid \exists f \in F \ i \xrightarrow[G]{u} f\}$. A Σ -word language is *regular* if it is recognized by a finite Σ -automaton. The class of regular Σ -word languages is a Boolean algebra and is denoted by $\text{Reg}(\Sigma^*)$.

2.4 Traces

2.4.1 Generalities

A *dependence relation* D is a reflexive and symmetric binary relation on Σ . The pair (Σ, D) is called a *dependence alphabet*. The complement of D is the *independence relation* $I := \Sigma^2 \setminus D$. The (Σ, D) -trace equivalence \equiv_D is the least congruence on Σ^* such that $(a, b) \in I \Rightarrow ab \equiv_D ba$. The (Σ, D) -trace of a word $w \in \Sigma^*$ is its \equiv_D -equivalence class and is denoted $[w]$. The quotient monoid Σ^* / \equiv_D is called the *trace monoid* of the dependence alphabet (Σ, D) and is denoted by $M(\Sigma, D)$. Note that in case of $D = \Sigma^2$, the trace monoid $M(\Sigma, D)$ coincides to the free monoid Σ^* .

The prefix binary relation \sqsubseteq on $M(\Sigma, D)$ defined by $t \sqsubseteq t'$ if and only if there exists $s \in M(\Sigma, D)$ such that $ts = t'$ is a partial ordering.

Consider the finite alphabet $I_D := \{A \subseteq \Sigma \mid \forall a_1 \neq a_2 \in A \ (a_1, a_2) \in I\}$, and denote by $\Pi_{I_D} : I_D^* \rightarrow M(\Sigma, D)$ the canonical morphism defined by $\Pi_{I_D}(\emptyset) = [\varepsilon]$ and $\Pi_{I_D}(\{a_1, \dots, a_n\}) = [a_1 \dots a_n]$ ($n \geq 0$). Given $P \subseteq I_D$, we denote by Π_P the restriction of Π_{I_D} to P^* . A P -word U encodes the trace $\Pi_P(U)$.

Consider the binary relation \triangleright on $I_D^- := I_D \setminus \{\emptyset\}$ defined by: $A \triangleright B \iff \forall b \in B \exists a \in A \ aDb$. Denote by $\mathbf{F} \subseteq I_D^{-*}$ the set of \triangleright -paths.

The surjective morphism $\Pi_{I_D^-}$ is not injective. Indeed, suppose $\Sigma = \{a, b\}$ and aIb , then $\Pi_{I_D^-}(\{a, b\}) = \Pi_{I_D^-}(\{a\}\{b\})$. The following proposition expresses that each trace is encodable by a unique I_D^- -word in \mathbf{F} .

Proposition 2.1 (Foata normal form). *Let $t \in M(\Sigma, D)$. The Foata normal form of t , $[t]_{\mathbf{F}}$, is the unique I_D^- -word $[t]_{\mathbf{F}} = A_1 \cdots A_p \in \mathbf{F}$ ($p \geq 0$) such that $\Pi_{I_D^-}(A_1 \cdots A_p) = t$.*

Example 2.2. Suppose $\Sigma = \{a, b, c, d\}$ and aIc, bId, cId . The Foata normal form of $t = [acbdab]$ (see Figure 1) is $[t]_{\mathbf{F}} = \{a, c\}\{b, d\}\{a\}\{b\}$.

Lemma 2.3 (Level automata). *The set \mathbf{F} of Foata normal forms is regular.*

$$\begin{array}{c}
 a \quad d \\
 \swarrow \quad \searrow \\
 c \quad b \\
 \searrow \quad \swarrow \\
 a \quad b
 \end{array}$$

Figure 1: The Foata normal form of $[acbdab]$ (Example 2.2)

Proof. It is recognized by the following finite I_D^- -automaton \mathcal{A}_F .

- The I_D^- -graph is given by:

$$\perp \xrightarrow{A} A : A \in I_D^-$$

$$A \xrightarrow{B} B : A \triangleright B$$

- the initial state is $\perp \notin I_D^-$
- all the states are final (even \perp).

□

2.4.2 Recognizable trace languages

A (Σ, D) -trace language is a subset of $M(\Sigma, D)$. If \mathcal{L} is a trace language, then the word language $\cup \mathcal{L}$ is $\cup \mathcal{L} = \{w \in \Sigma^* \mid [w] \in \mathcal{L}\}$. If L is a word language, then $[L]$ is the trace language defined by $[L] := \{[w] \in M(\Sigma, D) \mid w \in L\}$.

A trace language $\mathcal{L} \subseteq M(\Sigma, D)$ is *recognizable* if there exists a finite monoid N and a monoid morphism $\phi : M(\Sigma, D) \rightarrow N$ such that $\mathcal{L} = \phi^{-1}(\phi(\mathcal{L}))$. The class of recognizable trace languages is denoted by $\text{Rec}(M(\Sigma, D))$.

Remark 2.4. In case of $D = \Sigma^2$, $\text{Rec}(M(\Sigma, D)) = \text{Reg}(\Sigma^*)$.

The next proposition recalls the robustness of the class $\text{Rec}(M(\Sigma, D))$.

Proposition 2.5. $\text{Rec}(M(\Sigma, D))$ is a Boolean algebra closed under concatenation.

We give two characterizations of the recognizability of a trace language. The residual by $s \in M(\Sigma, D)$ of $\mathcal{L} \subseteq M(\Sigma, D)$ is $s^{-1}\mathcal{L} = \{t \in M(\Sigma, D) \mid s \cdot t \in \mathcal{L}\}$. For example, suppose $\Sigma = \{a, b\}$ and alb , then consider $\mathcal{L} = \{[ab], [abaa], [aaa], [aabb]\}$. The residual by $[ab]$ of \mathcal{L} is $\{\varepsilon, [aa], [abb]\}$. The recognizability of a trace language \mathcal{L} is characterized by the finiteness of its set of residuals.

Proposition 2.6. $\mathcal{L} \in \text{Rec}(M(\Sigma, D))$ if and only if $\{s^{-1}\mathcal{L} \mid s \in M(\Sigma, D)\}$ is finite.

Suppose P is a finite alphabet and $\pi : P^* \rightarrow M(\Sigma, D)$ is a surjective morphism. For instance, P could be Σ, I_D^- or I_D . If for a trace t we think of $\pi^{-1}(t)$ as the set of its P -encodings, the following proposition says that the recognizability of a trace language is equivalent to the regularity of the set of all P -encodings of its elements.

Proposition 2.7. $\mathcal{L} \in \text{Rec}(M(\Sigma, D))$ if and only if $\pi^{-1}(\mathcal{L})$ is regular.

3 Recognizable trace rewriting system with level-regular contexts

The trace language $[(ab)^*]$ with alb is not recognizable since it has an infinite set of residuals. Nevertheless, the set of Foata normal forms of its elements $\{a, b\}^*$ is regular. This suggests to consider a weaker form of recognizability. In this section, we define the notion of level-regularity for trace languages. Then we consider recognizable trace rewriting systems with level-regular contexts and we prove that their rewriting graphs are word-automatic.

Let (Σ, D) be a dependence alphabet. In the following, we write $\Pi_{\mathbf{F}}$ for the restriction of $\Pi_{I_D^-}$ to \mathbf{F} .

3.1 Level-regularity

Definition 3.1. $\mathcal{L} \subseteq M(\Sigma, D)$ is level-regular if the word language $\Pi_{\mathbf{F}}^{-1}(\mathcal{L})$ is regular.

By Proposition 2.7 and Lemma 2.3, every recognizable trace language is level-regular. Indeed, $\Pi_{\mathbf{F}}^{-1}(\mathcal{L}) = \Pi_{I_b}^{-1}(\mathcal{L}) \cap \mathbf{F}$.

The class of level-regular languages is a Boolean algebra but it is not closed under concatenation. Consider for example the concatenation of the two level-regular trace languages $[(ab)^*]$ and $[(bc)^*]$, with $D = \{(a, a), (b, b), (c, c)\}$. The set of Foata normal forms of its elements

$$\begin{aligned} & \Pi_{\mathbf{F}}^{-1}([(ab)^*] \cdot [(bc)^*]) \\ = & \{ \{a, b, c\}^k \{b, c\}^* \{b\}^k \mid k \geq 0 \} \cup \{ \{a, b, c\}^k \{a, b\}^* \{b\}^k \mid k \geq 0 \} \end{aligned}$$

is not regular.

3.2 Trace rewriting system

Graphs at the first level of the pushdown hierarchy are the suffix rewriting graphs of recognizable word rewriting systems. Such a rewriting system is a finite set of rules of the form $U \cdot (V \rightarrow W)$, where U (the context language), V and W are regular languages. In the following, we consider recognizable trace rewriting systems with level-regular contexts and recognizable left and right hand sides and we prove that their rewriting graphs are word-automatic by encoding their vertex sets by their Foata normal forms.

Definition 3.2. A recognizable trace rewriting system with level-regular contexts (RTL) \mathbf{R} on $M(\Sigma, D)$ is a finite set of rules of the form

$$\mathcal{U} \cdot (\mathcal{V} \xrightarrow{\lambda} \mathcal{W})$$

where \mathcal{U} is level-regular, $\mathcal{V}, \mathcal{W} \in \text{Rec}(M(\Sigma, D))$ and $\lambda \in \Lambda$ a set of labels.

The rewriting graph $\text{Gr}_{\mathbf{R}}$ of the RTL \mathbf{R} is the Λ -graph on $M(\Sigma, D)$ defined by

$$\text{Gr}_{\mathbf{R}} = \{ [uv] \xrightarrow{\lambda} [uw] \mid \exists \mathcal{U} \cdot (\mathcal{V} \xrightarrow{\lambda} \mathcal{W}) \in \mathbf{R}, [u] \in \mathcal{U}, [v] \in \mathcal{V}, [w] \in \mathcal{W} \}.$$

Example 3.3. Suppose $D = \{(a, a), (b, b)\}$ and consider the following RTL:

$$[(a+b)^*] \cdot ([\varepsilon] \xrightarrow{a} [a])$$

$$[(a+b)^*] \cdot ([\varepsilon] \xrightarrow{b} [b])$$

$$[(ab)^*] \cdot ([\varepsilon] \xrightarrow{f} [\varepsilon])$$

Its rewriting graph is the infinite grid with a loop labelled by f on each vertex of its diagonal (see Figure 2).

Example 3.4. Suppose $D = \{(a, a), (b, b), (c, c)\}$ and consider the following RTL \mathbf{R} :

$$[(abc)^*] \cdot ([\varepsilon] \xrightarrow{a} [abc])$$

$$[(abc)^*(ac)^*] \cdot ([b] \xrightarrow{b} [\varepsilon])$$

$$[(abc)^*(ac)^*] \cdot ([ac] \xrightarrow{c} [\varepsilon])$$

The rewriting graph $\text{Gr}_{\mathbf{R}}$ (see Figure 3) is not in the pushdown hierarchy because its MSO theory is undecidable. Furthermore, remark that without the c -inner edges, we obtain a graph belonging to level 2 of the pushdown hierarchy.

Before stating the main result (Theorem 3.8) of this section, we recall some basic definitions about word-automatic graphs.

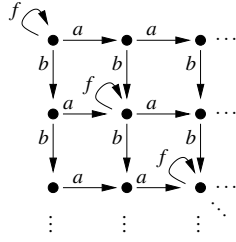


Figure 2: The diagonal of the infinite grid (Example 3.3)

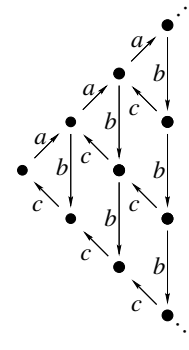


Figure 3: The rewriting graph of a RTL (Example 3.4)

Word-automatic graphs. Let Σ be an alphabet and $\# \notin \Sigma$ a new symbol. The synchronization of two Σ -words, $u = a_1 \dots a_m$ and $v = b_1 \dots b_n$, is the $(\Sigma \cup \{\#\})^2$ -word $u \otimes v$ defined by $u \otimes v := (a_1, b_1) \dots (a_k, b_k)(x_{k+1}, y_{k+1}) \dots (x_K, y_K)$, where $k = \min(m, n)$, $K = \max(m, n)$ and for every $k < i \leq K$, $(x_i, y_i) = (\#, b_i)$ if $k = m$ and $(x_i, y_i) = (a_i, \#)$ if not.

A Λ -graph (Λ a finite alphabet) G is *word-automatic* if there exists a regular word language L_{V_G} and a bijection $v : L_{V_G} \rightarrow V_G$ such that for each $\lambda \in \Lambda$, the synchronized word language $L_\lambda = \{v^{-1}(s) \otimes v^{-1}(t) \mid s \xrightarrow[\Lambda]{\lambda} t\}$ is regular.

The following proposition recalls that the domain and the image of any word-automatic relation is regular.

Lemma 3.5. *If a language L of $(\Sigma \cup \{\#\})^2$ -words $u \otimes v$ is regular, then the languages $\{u \in \Sigma^* \mid \exists v \in \Sigma^* u \otimes v \in L\}$ and $\{v \in \Sigma^* \mid \exists u \in \Sigma^* u \otimes v \in L\}$ are regular.*

Remark 3.6. By Lemma 3.5, a Λ -graph is word-automatic if and only if there exists a bijection $v : L \rightarrow V$, where $L \in \text{Reg}(\Sigma^*)$ and $V \supseteq V_G$ such that for each $\lambda \in \Lambda$, the $(\Sigma \cup \{\#\})^2$ -word language $L_\lambda = \{v^{-1}(s) \otimes v^{-1}(t) \mid s \xrightarrow[\Lambda]{\lambda} t\}$ is regular.

Remark 3.7. Let L be a regular P -word language. Then the $(P \cup \{\#\})^2$ -word language $\{u \otimes v \mid u, v \in L\}$ is regular. In particular, the $(I_D^- \cup \{\#\})^2$ -word language $\{[s]_{\mathbf{F}} \otimes [t]_{\mathbf{F}} \mid s, t \in M(\Sigma, D)\}$ is regular.

The following theorem is partially due to the unique encoding of any trace by its Foata normal form.

Theorem 3.8. *The rewriting graph of a recognizable trace rewriting with level-regular contexts (RTL graph) is word-automatic.*

Theorem 3.8 is no more guaranteed if we suppose that left and right hand sides are just level-regular (see Remark 3.12).

Corollary 3.9 ([8]). *The FO theory of a RTL graph is decidable.*

In order to prove Theorem 3.8, we set out a crucial property about compatibility between concatenation and Foata normal forms.

In general, $[st]_{\mathbf{F}} \neq [s]_{\mathbf{F}}[t]_{\mathbf{F}}$. Indeed, suppose $D = \{(a, a), (b, b)\}$. If $s = [a]$ and $t = [ab]$, then $[s]_{\mathbf{F}} = \{a\}$, $[t]_{\mathbf{F}} = \{a, b\}$ and $[st]_{\mathbf{F}} = \{a, b\}\{a\}$. The following lemma expresses some compatibility between concatenation and Foata normal form.

Lemma 3.10. *Let $s, t \in M(\Sigma, D)$ such that $[s]_{\mathbf{F}} = A_1 \dots A_p$ ($p \geq 0$) and $[st]_{\mathbf{F}} = B_1 \dots B_m$. Then $m \geq p$, $A_i \subseteq B_i$ for each $1 \leq i \leq p$ and $\Pi_{I_D}((B_1 \setminus A_1) \dots (B_p \setminus A_p) B_{p+1} \dots B_m) = t$.*

Proof. By induction on the length of t . □

In the following, for $\lceil s \rceil_{\mathbf{F}} = A_1 \cdots A_p$ ($p \geq 0$) and $t \in M(\Sigma, D)$, denote by $\lceil s \rceil_{\mathbf{F}} \parallel t$ the I_D -word language $B_1 \cdots B_m$ ($m \geq p$) such that $A_i \subseteq B_i$ for each $1 \leq i \leq p$ and $\Pi_{I_D}((B_1 \setminus A_1) \cdots (B_p \setminus A_p) B_{p+1} \cdots B_m) = t$. Thus $\lceil st \rceil_{\mathbf{F}} \in \lceil s \rceil_{\mathbf{F}} \parallel t$, by the lemma above.

Example 3.11. Suppose $D = \{(a, a), (b, b)\}$ and consider $s = [aba]$ and $t = [ab]$. Then $\lceil s \rceil_{\mathbf{F}} = \{a, b\}\{a\}$ and

$$\lceil s \rceil_{\mathbf{F}} \parallel t = \{a, b\}\{a\} \emptyset^* (\{a\} \emptyset^* \{b\} + \{b\} \emptyset^* \{a\} + \{a, b\}) \emptyset^* \cup \{a, b\}\{a, b\} \emptyset^* \{a\} \emptyset^*$$

Proof of Theorem 3.8. By Proposition 2.1 and Lemma 2.3, $\Pi_{\mathbf{F}}$ is a bijection from the regular language \mathbf{F} onto $M(\Sigma, D) \supseteq V_{\text{GrR}}$. We are going to prove that for each $\lambda \in \Lambda$, the $(I_D \cup \{\#\})^2$ -word language $L_\lambda = \{ \lceil [u][v] \rceil_{\mathbf{F}} \otimes \lceil [u][w] \rceil_{\mathbf{F}} \mid [u] \in \mathcal{U}, [v] \in \mathcal{V}, [w] \in \mathcal{W}, \mathcal{U} \cdot (\mathcal{V} \xrightarrow{\lambda} \mathcal{W}) \in \mathbf{R} \}$ is regular.

Let $\mathcal{U} \cdot (\mathcal{V} \xrightarrow{\lambda} \mathcal{W})$ be a rule in \mathbf{R} . We have to prove that the $(I_D \cup \{\#\})^2$ -word language $\{ \lceil [u][v] \rceil_{\mathbf{F}} \otimes \lceil [u][w] \rceil_{\mathbf{F}} \mid [u] \in \mathcal{U}, [v] \in \mathcal{V}, [w] \in \mathcal{W} \}$ is regular. By Lemma 3.10 and Remark 3.7 and because the intersection of two regular word languages is regular, it suffices to show that the language of $(I_D \cup \{\#\})^2$ -words of the form $X \otimes Y$ such that there exists $[u] \in \mathcal{U}, [v] \in \mathcal{V}$ and $[w] \in \mathcal{W}$ such that $X \in \lceil [u] \rceil_{\mathbf{F}} \parallel [v]$ and $Y \in \lceil [u] \rceil_{\mathbf{F}} \parallel [w]$, is regular. For this, consider the I_D -automata $\mathcal{A}_1, \mathcal{A}_2$ et \mathcal{A}_3 that recognize respectively $\{ \lceil [u] \rceil_{\mathbf{F}} \mid [u] \in \mathcal{U} \}$, $\Pi_{I_D}^{-1}(\mathcal{V})$ and $\Pi_{I_D}^{-1}(\mathcal{W})$ and define the following $(I_D \cup \{\#\})^2$ -automaton.

- The initial state is $(i_{\mathcal{A}_1}, i_{\mathcal{A}_2}, i_{\mathcal{A}_3})$
- the $(I_D \cup \{\#\})^2$ -graph is given by

$$(p, q, r) \xrightarrow{A \cup B / A \cup C} (p', q', r') : p \xrightarrow{\mathcal{A}_1} p', q \xrightarrow{\mathcal{A}_2} q', r \xrightarrow{\mathcal{A}_3} r'$$

$$(p, q, r) \xrightarrow{B/C} (\perp, q', r') : p \in F_{\mathcal{A}_1} \cup \{\perp\}, q \xrightarrow{\mathcal{A}_2} q', r \xrightarrow{\mathcal{A}_3} r'$$

$$(p, q, r) \xrightarrow{\# / C} (\perp, \perp, r') : p \in F_{\mathcal{A}_1} \cup \{\perp\}, q \in F_{\mathcal{A}_2}, r \xrightarrow{\mathcal{A}_3} r'$$

$$(\perp, \perp, r) \xrightarrow{\# / C} (\perp, \perp, r') : r \xrightarrow{\mathcal{A}_3} r'$$

$$(p, q, r) \xrightarrow{B / \#} (\perp, q', \perp) : p \in F_{\mathcal{A}_1} \cup \{\perp\}, r \in F_{\mathcal{A}_3}, q \xrightarrow{\mathcal{A}_2} q'$$

$$(\perp, q, \perp) \xrightarrow{B / \#} (\perp, q', \perp) : q \xrightarrow{\mathcal{A}_2} q'$$
- the set of final states is $F = \{(p, q, r) \mid p \in F_{\mathcal{A}_1} \cup \{\perp\}, q \in F_{\mathcal{A}_2}, r \in F_{\mathcal{A}_3}\} \cup \{(\perp, \perp, r) \mid r \in F_{\mathcal{A}_3}\} \cup \{(\perp, q, \perp) \mid q \in F_{\mathcal{A}_2}\}$.

□

Remark 3.12. Suppose $D = \{(a, a), (b, b), (c, c)\}$ and consider the following rewriting rule: $[(ab)^*][\varepsilon] \rightarrow [(bc)^*]$. Observe that $[(ab)^*]$ and $[(bc)^*]$ are level-regular but not recognizable. Recall that if a relation is word-automatic, then its image is regular (Proposition 3.5). The rewriting graph of this rewriting rule fails to be word-automatic by encoding its vertex set by their Foata normal forms because $\Pi_{\mathbf{F}}^{-1}([(ab)^*] \cdot [(bc)^*])$ is not regular.

The FO[Reach] theory of a RTL graph may fail to be decidable. Indeed, the halting problem of 2-counter Minsky machines can be encoded by RTL graphs.

Proposition 3.13. *There exists some RTL graphs that does not have a decidable FO[Reach] theory.*

Before proving the proposition above, let us recall some basic definitions about 2-counter Minsky machines.

A 2-counter Minsky machine M of length n is a sequence of n instructions. The n -th instruction is a special instruction that halts the machine and for each $k \in \{1, \dots, n-1\}$ the k -th instruction is of the form

$$\mid k : c := c + 1; \text{goto}(j) \quad (\text{Incr}(c, j))$$

or

$$\mid k : \text{if } c \neq 0 \text{ then } c := c - 1; \text{goto}(j) \text{ else } \text{goto}(l) \quad (\text{Decr}(c, j, l))$$

where $j, l \in \{1, \dots, n\}$ and c is one of the 2 counters.

Configurations of M are the triples $(k, c_1, c_2) \in \{1, \dots, n\} \times \mathbb{N} \times \mathbb{N}$, where k is the instruction number, and c_1 and c_2 the 2-counter contents. The initial configuration is $(1, 0, 0)$. A computation is a sequence of configurations starting from the initial configuration and such that two successive configurations respect the instructions. The halting problem is: given a 2-counter Minsky machine, is there a finite computation that halts the machine ?

Theorem 3.14 (Minsky). *The halting problem of 2-counter Minsky machines is undecidable.*

Proof of Proposition 3.13. Given a 2-counter Minsky machine M of length n , consider the rewriting graph G_M of the following recognizable trace rewriting system:

- $\Sigma := \{\perp_a, \perp_b, a, b, 1, \dots, n\}$
- the independence relation I on Σ is given by: $alb, \perp_a I \perp_b$
- for each $k \in \{1, \dots, n-1\}$ the rewriting rules are:
 - $k \xrightarrow{R} cj$ ($j \in \{1, \dots, n\}, c \in \{a, b\}$) if the k -th instruction is $\text{Incr}(c, j)$
 - $ck \xrightarrow{R} j$
 - and
 - $\perp_c k \xrightarrow{R} \perp_c l$ ($j, l \in \{1, \dots, n\}, c \in \{a, b\}$) if the k -th instruction is $\text{Decr}(c, j, l)$
 - $[\perp_a \perp_b 1]([\varepsilon] \xrightarrow{i} [\varepsilon])$
 - $[n][[\varepsilon] \xrightarrow{f} [\varepsilon])$

The initial configuration is encoded by $[\perp_a \perp_b 1]$. Final configurations are encoded by $[\perp_a \perp_b a^* b^* n]$.

A configuration (k, c_1, c_2) accessible from $[\perp_a \perp_b 1]$ is encoded by the trace $[\perp_a \perp_b \overbrace{a \dots a}^{c_1} \overbrace{b \dots b}^{c_2} k]$.

The machine M halts if and only if G_M satisfies: $\exists x \exists y (x \xrightarrow{i} x \wedge y \xrightarrow{f} y \wedge x \xrightarrow{*} y)$.

□

4 Concurrent unfolding of a concurrent automaton

In this section, we consider concurrent automata, that were first introduced in [13] as asynchronous transition systems, and we prove that the FO[Reach] theory of their concurrent unfoldings is decidable. Indeed, we will show that the concurrent unfolding of a concurrent automaton, with the reachability relation is a RTL graph.

Let (Σ, D) be a dependence alphabet and $I = \Sigma^2 \setminus D$.

$$D = \{(a, a), (b, b)\}$$

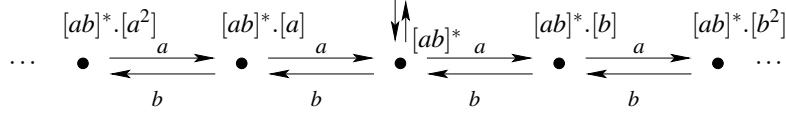


Figure 4: $\text{Res}([(ab)^*], \Sigma)$ (Example 4.2)

4.1 Concurrent automata

Definition 4.1. An Σ -automaton $\mathcal{A} = (G, i, F)$ is D -concurrent when

- G is deterministic
- $((a, b) \in I \text{ and } p \xrightarrow[\mathcal{A}]{ab} q) \implies p \xrightarrow[\mathcal{A}]{ba} q$.

Every automaton can be seen as a concurrent automaton relatively to the total dependence relation on its edge label set.

Example 4.2. Let $\mathcal{L} \subseteq M(\Sigma, D)$ be a trace language. The residual automaton of \mathcal{L} by Σ is the D -concurrent Σ -automaton $\text{Res}(\mathcal{L}, \Sigma)$ defined by:

- the Σ -graph $\{[u]^{-1} \mathcal{L} \xrightarrow{a} [ua]^{-1} \mathcal{L} \mid u \in \Sigma^*, a \in \Sigma\}$
- the initial state \mathcal{L}
- final states $[u]^{-1} \mathcal{L}$ such that $[\varepsilon] \in [u]^{-1} \mathcal{L}$,

is a D -concurrent Σ -automaton that recognises $\cup \mathcal{L}$ (see Figure 4).

Example 4.3. Let $\mathcal{L} \subseteq M(\Sigma, D)$ be a trace language. The unfolding automaton $U(\mathcal{L}, \Sigma)$ of \mathcal{L} by Σ defined by

- the Cayley graph of $M(\Sigma, D)$: $\{[u] \xrightarrow{a} [ua] \mid u \in \Sigma^*, a \in \Sigma\}$
- the initial state $[\varepsilon]$
- final states $t \in \mathcal{L}$

is a D -concurrent Σ -automaton that recognises $\cup \mathcal{L}$.

By combining Proposition 2.7 and Example 4.2, we obtain the following characterization of recognizable trace languages:

Proposition 4.4. A trace language \mathcal{L} is recognizable if and only if there exists a finite D -concurrent Σ -automaton \mathcal{A} such that $\cup \mathcal{L} = L(\mathcal{A})$.

4.2 The concurrent unfolding of a concurrent automaton

Definition 4.5. The D -unfolding $\text{Unf}_D(\mathcal{A})$ of a D -concurrent Σ -automaton \mathcal{A} is the D -concurrent $\Sigma \cup \{f\}$ -graph defined by:

$$\text{Unf}_D(\mathcal{A}) = \{[u] \xrightarrow{a} [ua] \mid u \in \Sigma^*, a \in \Sigma, i \xrightarrow[\mathcal{A}]{ua}\} \cup \{[u] \xrightarrow{f} [u] \mid u \in L(\mathcal{A})\}.$$

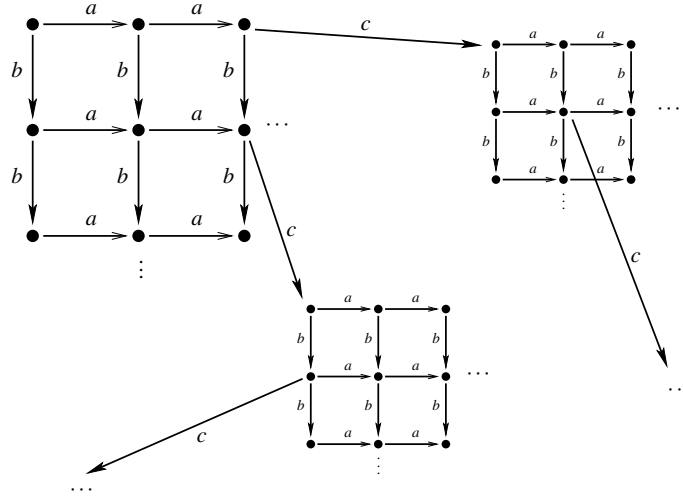


Figure 5: The infinite grid tree

In the following example, we introduce the infinite grid tree as the concurrent unfolding of a finite concurrent automaton.

Example 4.6. Let $\Sigma = \{a, b, c\}$ and suppose alb . Consider the graph $G = \{p \xrightarrow{a,b,c} p\}$. The D -unfolding of $\mathcal{A} = (G, p, \emptyset)$ (Figure 5), is the infinite grid (on $\{a, b\}$) tree (see Section 5) and has a decidable FO[Reach] theory by the theorem below.

Before stating the main result of this section, recall that the unfolding of a finite graph is a regular tree whose monadic second-order theory is decidable (since unfolding preserves monadic second-order decidability). Here, we consider a notion of concurrent unfolding and we apply this graph transformation to a wider class than the class of finite graphs.

Theorem 4.7. *If \mathcal{A} is a finite D -concurrent automaton, then the FO[Reach] theory of $\text{Unf}_D(\mathcal{A})$ is decidable.*

We do not know if, in general, the D -unfolding preserves FO[Reach] decidability.

Proof of Theorem 4.7. Consider the $\Sigma \dot{\cup} \{*\}$ -automaton

$$\text{Unf}_D(\mathcal{A})_* := \text{Unf}_D(\mathcal{A}) \cup \{[u] \xrightarrow{*} [uv] \mid u, v \in \Sigma^*, i \xrightarrow{\mathcal{A}} uv\}$$

It is the rewriting graph of the following recognizable trace rewriting system:

$$\left\{ \begin{array}{l} [L(G, i, Q_a)]([\varepsilon] \xrightarrow{a} [a]) \quad a \in \Sigma \text{ and } Q_a = \{q \in Q \mid q \xrightarrow{a} \} \\ [L(G, i, F)]([\varepsilon] \xrightarrow{f} [\varepsilon]) \\ [L(G, i, q)]([\varepsilon] \xrightarrow{*} [L(G, q, Q)]) \quad q \in Q. \end{array} \right.$$

□

Remark 4.8. Given a Σ -graph G , the FO theory of the graph $G \cup \{p \xrightarrow{\mathcal{L}} q \mid p \xrightarrow{u} q, [u] \in \mathcal{L}, \mathcal{L} \in \text{Rec}(M(\Sigma, D))\}$ is referred to as the first-order theory with recognizable reachability predicates (FO[Rec]) of G . We can strengthen the last theorem and show that: if \mathcal{A} is a finite D -concurrent automaton, then $\text{Unf}_D(\mathcal{A})$ has a decidable FO[Rec] theory. Indeed, observe that each sentence in FO[Rec] logic contains

a finite number of atomic formula $x \xrightarrow{\mathcal{L}_1} y, \dots, x \xrightarrow{\mathcal{L}_n} y$ ($n \geq 1$). Then $\text{Unf}_D(\mathcal{A}) \cup \{p \xrightarrow{\mathcal{L}_j} q \mid p \xrightarrow{u} q, [u] \in \mathcal{L}_j, j \in \{1, \dots, n\}\}$ is the rewriting graph of the following RTL:

$$\begin{cases} [L(G, i, Q_a)]([\varepsilon] \xrightarrow{a} [a]) & a \in \Sigma \text{ and } Q_a = \{q \in Q \mid q \xrightarrow{\mathcal{A}}\} \\ [L(G, i, F)]([\varepsilon] \xrightarrow{f} [\varepsilon]) \\ [L(G, i, q)]([\varepsilon] \xrightarrow{\mathcal{L}_j} [L(G, q, Q)] \cap \mathcal{L}_j) & q \in Q \ j \in \{1, \dots, n\}. \end{cases}$$

We have deduced the FO[Rec] theory decidability of the Cayley graph of a trace monoid from the FO decidability of RTL graphs. The following remark shows the inverse reduction.

Remark 4.9. Lastly, note that any rewriting graph of a recognizable trace rewriting system (with recognizable contexts) on some trace monoid $M(\Sigma, D)$ is a FO[Rec] interpretation of the Cayley graph of this trace monoid. Indeed, observe that the neutral element is FO-definable: $\text{neutral}(x) = \forall t \bigwedge_{a \in \Sigma} \neg(t \xrightarrow{a} x)$. Then for each rule of the form $\mathcal{U} \cdot (\mathcal{V} \rightarrow \mathcal{W})$ consider the formula: $\phi(x, y) = \exists i \exists z (\text{neutral}(i) \wedge i \xrightarrow{\mathcal{U}} z \wedge z \xrightarrow{\mathcal{V}} x \wedge z \xrightarrow{\mathcal{W}} y)$.

4.3 Regular trace event structure

In [11], Madhusudan prove that the FO theory of a regular trace event structure is decidable. For this, he shows that the vertex set and the relations of such a graph can be encoded by a recognizable trace language on a judicious dependence alphabet. Note that, due to the level-regular contexts, this technique does not allow to prove that the FO theory of RTL graphs is decidable.

A trace $t = [a_1 \cdots a_n] \in M(\Sigma, D)$ is *prime* if the set $\{1, \dots, n\}$, partially ordered by the relation E defined by iEj if and only if $i < j$ and $a_i D a_j$, has exactly one maximal element.

Let $\mathcal{L} \subseteq M(\Sigma, D)$ be a trace language. Denote by $\text{prime}(\mathcal{L})$ the set of prime traces in \mathcal{L} .

Definition 4.10. The event structure defined by \mathcal{L} , $\mathcal{E}\mathcal{S}\mathcal{L}$, is the $\{\leq, \#, (\lambda_a)_{a \in \Sigma}\}$ -graph whose vertex set is $\text{prime}(\mathcal{L})$ defined by

- $t \xrightarrow{\leq} t' : t \sqsubseteq t'$
- $t \xrightarrow{\#} t' : \forall t'' \in \text{prime}(\mathcal{L}) (t \not\sqsubseteq t'' \vee t' \not\sqsubseteq t'')$
- $t \xrightarrow{\lambda_a} t : \text{the maximal element of } t \text{ is } a$.

Theorem 4.11 ([11]). *If $\mathcal{L} \in \text{Rec}(M(\Sigma, D))$, then $\mathcal{E}\mathcal{S}\mathcal{L}$ has a decidable FO theory.*

Proof. A trace $t \in V_{\text{Unf}_D(\text{Res}(\mathcal{L}, \Sigma))_*}$ is prime if and only if t is not successor of two distinct vertices of $\text{Unf}_D(\text{Res}(\mathcal{L}, \Sigma))_*$. The event structure $\mathcal{E}\mathcal{S}\mathcal{L}$ can be obtained by a FO interpretation of $\text{Unf}_D(\text{Res}(\mathcal{L}, \Sigma))_*$ that has a decidable FO theory. \square

5 Graph tree

In this section, we consider ground term rewriting graphs. These graphs have a decidable FO[Reach] theory [6]. We define a notion of graph tree and we prove that if a graph tree is a ground term rewriting graph (GTR graph), then it is finitely decomposable by size. A direct consequence is that the infinite grid tree, defined above as the concurrent unfolding of a concurrent automaton (Exemple 5), is not a GTR graph, although it has a FO[Reach] theory decidable.

5.1 Ground Term Rewriting graphs (GTR graphs)

A *position* is an element of \mathbb{N}^* , the set of finite words over \mathbb{N} . Denote by \sqsubseteq the prefix ordering over \mathbb{N}^* . Let F be a ranked alphabet (each symbol in F has an arity in \mathbb{N}). A *term* t on F is a partial function $t : \mathbb{N}^* \rightarrow F$ whose domain, $\text{Dom}(t)$, has the following properties :

- $\text{Dom}(t) \neq \emptyset$
- $\text{Dom}(t)$ is prefix closed
- $\forall u \in \text{Dom}(t)$, if the arity of $t(u)$ is n ($n \geq 0$), then $\{j \mid uj \in \text{Dom}(t)\} = \{1, \dots, n\}$.

The *size* $|t|$ of a term t is the number of its nodes. The *subterm* of t at position u , denoted $t \downarrow u$, is the term on F defined by :

- $\text{Dom}(t \downarrow u) = \{v \in \mathbb{N}^* \mid uv \in \text{Dom}(t)\}$
- $\forall v \in \text{Dom}(t \downarrow u)$, $(t \downarrow u)(v) = t(uv)$.

If $u \in \text{Dom}(t)$ and s is a term, then $t[u \leftarrow s]$, the term obtained from t by replacing the subterm $t \downarrow u$ by s , is defined by :

$$t[u \leftarrow s](v) = \begin{cases} s(w) & \text{if } v = uw \text{ and } w \in \text{Dom}(s) \\ t(v) & \text{if } v \in \text{Dom}(t) \text{ and } u \not\sqsubseteq v \end{cases}$$

If t is a term on F and $u \in \text{Dom}(t)$, then the *context of t at the position u* is the term $t[u \leftarrow x]$ on $F \cup \{x\}$, where x is a constant *i.e* the arity of x is 0.

A *context* C on F is a term on $F \cup \{x\}$, x constant, such that there exists a unique position $u_C \in \text{Dom}(C)$ for which $C(u_C) = x$. If t is a term on F , then the term $C[t]$ on F is defined by $C[t] := C[u_C \leftarrow t]$. The size $|C|$ of a context C on F is the number of its nodes minus 1.

A *ground term rewriting system* R is a 4-tuple $R = (F, \Sigma, R, i)$ where :

- F is a ranked alphabet
- Σ is a label alphabet
- $R := \bigcup_{a \in \Sigma} R_a$, where for each $a \in \Sigma$, R_a is a finite set of rules of the form $s \xrightarrow{a} s'$ with s and s' distinct terms on F
- i is an initial F -term.

We write :

- $t \xrightarrow[R]{a} t'$ if there exists a position $p \in \text{Dom}(t)$ and a rule $s \xrightarrow{a} s' \in R_a$ such that $t \downarrow p = s$ and $t' = t[p \leftarrow s']$
- $t \xrightarrow[R]{} t'$ when there exists $a \in \Sigma$ such that $t \xrightarrow[R]{a} t'$
- $\xrightarrow[R]^*$ for the reflexive and transitive closure under composition of $\xrightarrow[R]$.

The *configuration graph* Gr_R of R is the Σ -graph defined by

$$\text{Gr}_R := \{t \xrightarrow[R]{a} t' \mid i \xrightarrow[R]^* t, a \in \Sigma\}$$

A graph is called *Ground Term Rewriting graph (GTR graph)* if it is isomorphic to the configuration graph of a ground term rewriting system.

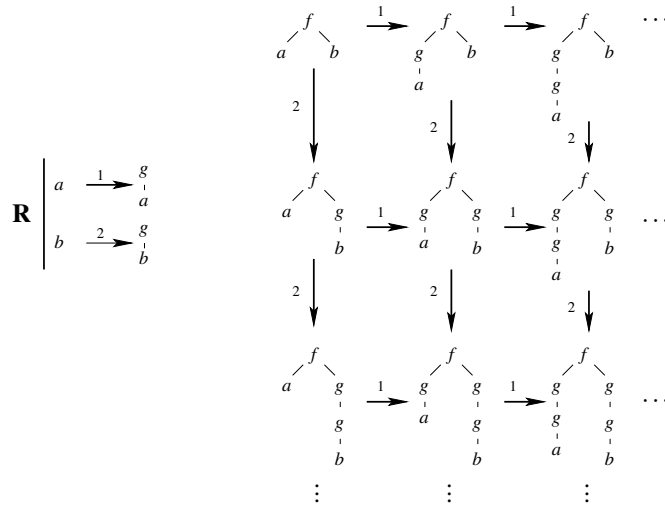


Figure 6: The infinite grid is a GTR graph

Remark 5.1. GTR graphs have no loop since each rule in the rewriting system has distinct left hand side and right hand side.

Example 5.2. The infinite grid is a GTR graph (see Figure 6).

In [6], Dauchet and Tison prove that a GTR graph with the reachability relation is tree-automatic. Thus:

Theorem 5.3 ([6]). *GTR graphs have a decidable FO[Reach] theory.*

5.2 Finite decomposition of a graph

Let us start by recalling the definition of the frontier of a subgraph.

Definition 5.4. Let G be a graph and $H \subseteq G$ a subgraph of G . The frontier of H (in G) is $\text{Fr}(H) = V_H \cap V_{G-H}$.

The frontier of H is the set of H -vertices that are incident to an edge in $G - H$.

Let Gr_R be a GTR graph. For each $n \geq 0$,

$$G_n := \{s \xrightarrow{e} t \in \text{Gr}_R \mid |s| < n \text{ or } |t| < n\}$$

According to Definition 5.4, the frontier of $\text{Gr}_R - G_n$ is $\text{Fr}(\text{Gr}_R - G_n) = V_{\text{Gr}_R - G_n} \cap V_{G_n}$. And, the frontier of K , a connected component of $\text{Gr}_R - G_n$, is $\text{Fr}(K) = \text{Fr}(\text{Gr}_R - G_n) \cap V_K$. The frontier of K are the K -vertices incident to an edge in G_n .

The graph Gr_R is *finitely decomposable by size* if

$$\text{dec} := \{(K, \text{Fr}(K)) \mid K \text{ connected component of } \text{Gr}_R - G_n, n \geq 0\}$$

has finite index, for the isomorphism relation.

Theorem 5.5 ([4]). *If a countable graph is finitely decomposable by size, then it is at the first level of the pushdown hierarchy. In particular, it has a decidable MSO theory.*

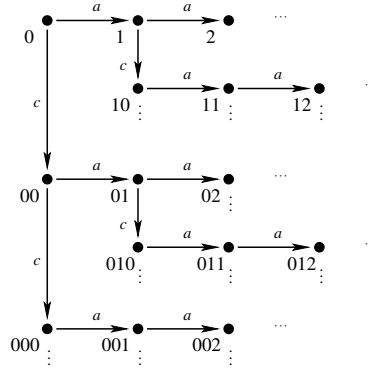


Figure 7: The semi-line tree

5.3 Graph tree and finite decomposition

Definition 5.6. Let G be a Σ -graph and $p_0 \in V_G$. Given a new symbol $c \notin \Sigma$, the G -tree from p_0 is the $\Sigma \cup \{c\}$ -graph, $\text{Tree}(G, p_0)$, defined by

$$\text{Tree}(G, p_0) := \{up \xrightarrow{a} uq \mid u \in V_G^*, p \xrightarrow{a} q\} \cup \{u \xrightarrow{c} up_0 \mid u \in V_G^*\}$$

Example 5.7. See Figure 7 for the semi-line tree.

Remark 5.8. The graph $\text{Tree}(G, p_0)$ is c -deterministic:

$$(v \xrightarrow[\text{Tree}(G, p_0)]{c} v_1 \text{ and } v \xrightarrow[\text{Tree}(G, p_0)]{c} v_2) \implies v_1 = v_2$$

Remark 5.9. The graph $\text{Tree}(G, p_0)$ is a tree if and only if G is a tree.

Remark 5.10. Let (Σ, D) be a dependence alphabet, G a finite D -concurrent Σ -graph and $p_0 \in V_G$. The $\Sigma \cup \{c\}$ -graph defined by $G \cup \{p \xrightarrow{c} p_0 \mid p \in V_G\}$ is D_c -concurrent, with $D_c = D \cup (\Sigma \cup \{c\}) \times (\Sigma \cup \{c\})$. Its D_c -unfolding from p_0 is $\text{Tree}(\text{Unf}_D(G, p_0), p_0)$.

Theorem 5.11. *If $\text{Tree}(G, p_0)$ is a GTR graph, then $\text{Tree}(G, p_0)$ is finitely decomposable by size.*

The MSO theory of the infinite grid is undecidable. The same holds for the infinite grid tree. By combining Theorem 5.11 and Theorem 5.5, we deduce the corollary below.

Corollary 5.12. *The infinite grid tree is not a GTR graph.*

5.4 Proof of Theorem 5.11

Lemma 5.13. *Let G be a Σ -graph and $p_0 \in V_G$. If there exists a ground term rewriting system $R = (F, \Sigma, R, i)$ such that $\text{Tree}(G, p_0)$ is isomorphic to Gr_R , then for every term $t \in V_{\text{Gr}_R}$, there exists a smallest position u_t (for the prefix ordering \sqsubseteq) at which t is incident to a rewriting in Gr_R .*

Proof of Lemma 5.13. It suffices to prove that if there exist two incomparable positions u' and u'' at which t is incident to rewritings, then there exists a position v , $v \sqsubseteq u'$, $v \sqsubseteq u''$ at which t is incident to a rewriting.

Denote by e' (respectively e'') the label of the rewriting t is incident in position u' (respectively u''). We are going to show that $c \notin \{e', e''\}$. Since u' and u'' are incomparable and because of Remark 5.1, there exists two paths between two distinct vertices of Gr_R , labelled by $e'e''$ and $e''e'$, each of them with no loop (see Figure 8). This is possible in $\text{Tree}(G, p_0)$ only if $c \notin \{e', e''\}$. Indeed,

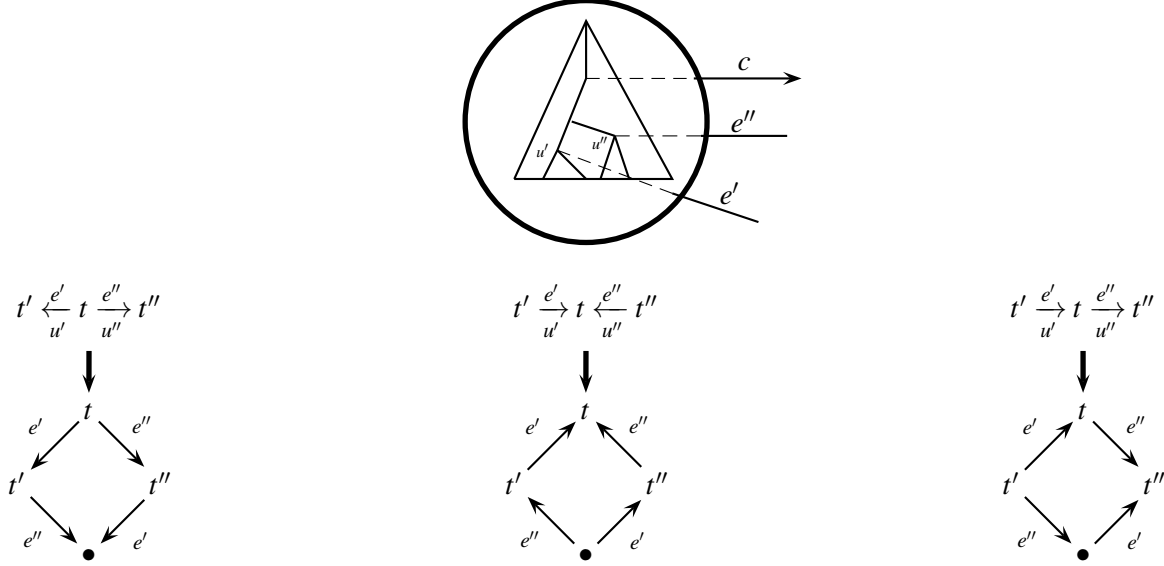


Figure 8: Paths in Gr_R between two distinct vertices, labelled by $e' e''$ and $e'' e'$

- $\{e', e''\} \subseteq \{c\}$ is impossible because of Remark 5.8
- $e' = c$ and $e'' \in \Sigma$ (or the converse $e'' = c$ and $e' \in \Sigma$) is impossible because c and e'' do not commute in $\text{Tree}(G, p_0)$.

But there exists a position v at which the term t is incident to a rewriting labelled by c . Due to the precedent point, the position v must be comparable to positions u' and u'' . Since u' and u'' are not comparable, we deduce that $v \sqsubseteq u'$ and $v \sqsubseteq u''$. \square

Proof of Theorem 5.11. Let $R = (F, \Sigma, R, i)$ be a ground term rewriting system such that $\text{Tree}(G, p_0)$ is isomorphic to Gr_R . We have to show that

$$\text{dec} := \{(K, V_K \cap V_{G_n}) \mid K \text{ connected component of } \text{Gr}_R - G_n, n \geq 0\}$$

has finite index. Let $\delta := \max\{|d| - |g| \mid g \xrightarrow{e} d \in R\}$ and $M := \max\{|g|, |d| \mid g \xrightarrow{e} d \in R\}$. We are going to show that for each connected component K in dec , there exists a position u_K and a context C_K such that

- for every term $t \in V_K$, $u_K \in \text{Dom}(t)$ and C_K is the context of t at the position u_K ($t = C_K[t \downarrow u_K]$)
- for every term $t \in \text{Fr}_{\text{Gr}_R}(K)$, $|t \downarrow u_K| < M + \delta$.

Then the finite subset of the (finite) set of terms whose size is at most $M + \delta$, obtained from $\text{Fr}_{\text{Gr}_R}(K)$ by removing the context C_K , is characteristic of the isomorphism type of $(K, \text{Fr}_{\text{Gr}_R}(K))$. Indeed, for $K \in \text{dec}$, let $\tilde{K} := \{s \mid C_K[s] \in \text{Fr}_{\text{Gr}_R}(K)\}$. If $\tilde{K} = \tilde{K}'$, then $(K, \text{Fr}_{\text{Gr}_R}(K))$ and $(K', \text{Fr}_{\text{Gr}_R}(K'))$ are isomorphic via $C_K[s] \mapsto C_{K'}[s]$.

Let $K \in \text{dec}$ and $n \geq 0$ such that K is a connected component of $\text{Gr}_R - G_n$. Remark that for every $t \in \text{Fr}_{\text{Gr}_R}(K)$, $n \leq |t| < n + \delta$. In particular, $\text{Fr}_{\text{Gr}_R}(K)$ is finite.

Let $m_K := \min\{|t| \mid t \in V_K\}$. Thus $n \leq m_K$. Consider $t_K \in V_K$ such that $|t_K| = m_K$. Since t_K is not an isolated vertex in K , there exists a position u at which t_K is incident to a rewriting in K . Let u_K be the

smallest prefix of u such that $|t_K \downarrow u_K| \leq M$. The term t_K can be written $t_K = C_K[t_K \downarrow u_K]$, with C_K a context.

We are going to prove that each term $t \in V_K$ is defined at position u_K and the context of t at u_K is C_K . It is sufficient to prove the following claim.

Claim 1. *Let $t \in V_K$. The position u_K is prefix of every position at which the term t is incident to a rewriting in K .*

Let $t \in \text{Fr}_{\text{GTR}}(K)$. Recall that $n \leq |t| < n + \delta$. Since $|t \downarrow u_K| = |t| - |C_K|$, we deduce $|t \downarrow u_K| < n + \delta - |C_K| \leq m_K + \delta - |C_K|$. But we have $m_K - |C_K| = |t_K \downarrow u_K| \leq M$. It follows that $|t \downarrow u_K| < M + \delta$. \square

Proof of Claim 1. First, let us show that the claim holds for the term t_K . Recall that there exists a position u at which t_K is incident to a rewriting in K and u_K is the smallest prefix of u such that $|t_K \downarrow u_K| \leq M$. The smallest position, u_{t_K} , at which the term t_K is incident to a rewriting (Lemma 5.13) is greater than u_K . Indeed, $u_{t_K} \sqsubseteq u$ and $|t_K \downarrow u_{t_K}| \leq M$. Therefore, every position v at which t_K is incident to a rewriting in K is such that: $u_K \sqsubseteq u_{t_K} \sqsubseteq v$.

Then, let $t \in V_K$ a term such that

- t is defined at position u_K
- the context of t at u_K is C_K
- the position u_K is prefix of every position at which t is incident to a rewriting in K

Consider a term t' and a position u at which the terms t and t' are incident to a same rewriting in K . Let us show that u_K is prefix of every position at which t' is incident to a rewriting in K . The smallest position, $u_{t'}$, at which the term t' is incident to a rewriting (Lemma 5.13) is not smaller than u_K . Indeed, we have $|t' \downarrow u_{t'}| \leq M$ and for every position v smaller than u_K , we have $|t' \downarrow v| > |t' \downarrow u_K| \geq |t_K \downarrow u_K|$. Thus, for the same reason as above for the term t_K , every position v at which t' is incident to a rewriting in K is greater than u_K .

Lastly, since K is connected, the claim is proved. \square

6 Conclusion

We have shown that a RTL graph is word-automatic and thus its first-order theory is decidable. We have also shown that such a graph does not have a decidable FO[Reach] theory. Furthermore, we have shown that the concurrent unfolding of a concurrent automaton with the reachability relation is a RTL graph and therefore its FO[Reach] theory is decidable. Lastly, we have shown that the class of concurrent unfoldings of finite concurrent automata is not included in the class of GTR graphs since the infinite grid tree is not a GTR graph.

Summing up, we have extended the first level of the pushdown hierarchy that consists of suffix rewriting graphs of recognizable word rewriting systems, to RTL graphs. Graphs at the first level of the pushdown hierarchy are the monadic interpretations of regular trees, that are concurrent unfoldings of finite concurrent automata for a trivial dependence relation. A RTL graph is FO[Rec] interpretation of the Cayley graph of the underlying trace monoid (Remark 4.9), that is the concurrent unfolding of a finite concurrent automaton. But we do not know whether reciprocally an FO[Rec] interpretation of a concurrent unfolding of a finite concurrent automaton is a RTL graph. We do not either know whether the concurrent unfolding transformation preserves FO[Reach] decidability. Another interesting problem would be to extend the second level of the pushdown hierarchy.

References

- [1] A. Carayol & S. Wöhrle (2003): *The Caucal Hierarchy of Infinite Graphs in Terms of Logic and Higher-Order Pushdown Automata*. In: *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, 23rd Conference, Mumbai, India, December 15-17, 2003, Proceedings*, pp. 112–123, doi:10.1007/978-3-540-24597-1_10. Available at https://doi.org/10.1007/978-3-540-24597-1_10.
- [2] D. Caucal (1990): *On the Regular Structure of Prefix Rewriting*. In: *CAAP '90, 15th Colloquium on Trees in Algebra and Programming, Copenhagen, Denmark, May 15-18, 1990, Proceedings*, pp. 87–102, doi:10.1007/3-540-52590-4_42. Available at https://doi.org/10.1007/3-540-52590-4_42.
- [3] D. Caucal (2002): *On Infinite Terms Having a Decidable Monadic Theory*. In: *Mathematical Foundations of Computer Science 2002, 27th International Symposium, MFCS 2002, Warsaw, Poland, August 26-30, 2002, Proceedings*, pp. 165–176, doi:10.1007/3-540-45687-2_13. Available at http://dx.doi.org/10.1007/3-540-45687-2_13.
- [4] D. Caucal (2008): *Deterministic graph grammars*. In: *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, pp. 169–250.
- [5] T. Colcombet & C. Löding (2007): *Transforming structures by set interpretations*. *Logical Methods in Computer Science* 3(2), doi:10.2168/LMCS-3(2:4)2007. Available at [https://doi.org/10.2168/LMCS-3\(2:4\)2007](https://doi.org/10.2168/LMCS-3(2:4)2007).
- [6] M. Dauchet & S. Tison (1990): *The Theory of Ground Rewrite Systems is Decidable*. In: *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pp. 242–248, doi:10.1109/LICS.1990.113750. Available at <http://dx.doi.org/10.1109/LICS.1990.113750>.
- [7] H. Ebbinghaus, J. Flum & W. Thomas (1996): *Mathematical Logic*. Undergraduate Texts in Mathematics, Springer New York. Available at <https://books.google.com/books?id=VYLA8m7cqYcC>.
- [8] B. Hodgson (1982): *On direct products of automaton decidable theories*. *Theoretical Computer Science* 19(3), pp. 331 – 335, doi:[https://doi.org/10.1016/0304-3975\(82\)90042-1](https://doi.org/10.1016/0304-3975(82)90042-1). Available at <http://www.sciencedirect.com/science/article/pii/0304397582900421>.
- [9] K. Lodaya (2007): *Petri Nets, Event Structures and Algebra*. In: *Formal Models, Languages and Applications [this volume commemorates the 75th birthday of Prof. Rani Siromoney]*, pp. 246–259, doi:10.1142/9789812773036_0017. Available at https://doi.org/10.1142/9789812773036_0017.
- [10] C. Löding (2003): *Infinite Graphs Generated by Tree Rewriting*. Ph.D. thesis, RWTH Aachen.
- [11] P. Madhusudan (2003): *Model-checking Trace Event Structures*. In: *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, pp. 371–380, doi:10.1109/LICS.2003.1210077. Available at <https://doi.org/10.1109/LICS.2003.1210077>.
- [12] S. Schulz (2010): *First-Order Logic with Reachability Predicates on Infinite Systems*. In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, pp. 493–504, doi:10.4230/LIPIcs.FSTTCS.2010.493. Available at <https://doi.org/10.4230/LIPIcs.FSTTCS.2010.493>.
- [13] M. Shields (1997): *Asynchronous Transition Systems*, pp. 183–189. Springer London, London, doi:10.1007/978-1-4471-0933-4_15. Available at https://doi.org/10.1007/978-1-4471-0933-4_15.
- [14] G. Winskel & M. Nielsen (1995): *Handbook of Logic in Computer Science (Vol. 4)*. chapter Models for Concurrency, Oxford University Press, Oxford, UK, pp. 1–148. Available at <http://dl.acm.org/citation.cfm?id=218623.218630>.
- [15] S. Wöhrle & W. Thomas (2004): *Model Checking Synchronized Products of Infinite Transition Systems*. In: *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*, pp. 2–11, doi:10.1109/LICS.2004.1319595. Available at <http://dx.doi.org/10.1109/LICS.2004.1319595>.