

Petri Net Reachability Graphs: Decidability Status of First Order Properties

Philippe Darondeau, Stephane Demri, Roland Meyer, Christophe Morvan

► **To cite this version:**

Philippe Darondeau, Stephane Demri, Roland Meyer, Christophe Morvan. Petri Net Reachability Graphs: Decidability Status of First Order Properties. Logical Methods in Computer Science, Logical Methods in Computer Science Association, 2012, 8 (4:9), pp.1-28. 10.2168/LMCS-8(4:9)2012 . hal-00743935

HAL Id: hal-00743935

<https://hal.inria.fr/hal-00743935>

Submitted on 22 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PETRI NET REACHABILITY GRAPHS: DECIDABILITY STATUS OF FIRST-ORDER PROPERTIES *

PHILIPPE DARONDEAU ^a, STEPHANE DEMRI ^b, ROLAND MEYER ^c,
AND CHRISTOPHE MORVAN ^d

^a INRIA Rennes Bretagne Atlantique, Campus de Beaulieu, Rennes, France
e-mail address: philippe.darondeau@inria.fr

^b LSV, ENS Cachan, CNRS, INRIA, France
e-mail address: Stephane.Demri@lsv.ens-cachan.fr

^c University of Kaiserslautern, Germany
e-mail address: meyer@cs.uni-kl.de

^d Université Paris-Est, Marne-La-Vallée, France
e-mail address: christophe.morvan@univ-paris-est.fr

ABSTRACT. We investigate the decidability and complexity status of model-checking problems on unlabelled reachability graphs of Petri nets by considering first-order and modal languages without labels on transitions or atomic propositions on markings. We consider several parameters to separate decidable problems from undecidable ones. Not only are we able to provide precise borders and a systematic analysis, but we also demonstrate the robustness of our proof techniques.

1. INTRODUCTION

Decision problems for Petri nets. Petri nets are among the oldest families of generators of infinite state systems, and much effort has been dedicated to their algorithmic analysis. For Petri nets, the reachability problem is hard but decidable [35]. Further important problems that are specific to Petri nets and that were shown decidable are boundedness [29, 38], deadlock-freeness and liveness [20] (by reduction to reachability), persistence [18], and semi-linearity [22]. Hack's thesis [20] provides a comprehensive overview of problems equivalent to Petri net reachability. On the negative side, language equality is undecidable for labelled Petri nets [21, 1], but it can be decided for injectively labelled as well as for labelled and deterministic Petri nets [37] (by a reduction to reachability). Another undecidability result for Petri nets, obtained by Rabin [4] and Hack [21], is that equality of reachability sets of two Petri nets with identical places is undecidable. As our main contribution, we link this result to first-order logic expressing properties of general Petri net reachability graphs.

1998 ACM Subject Classification: F.1.1, F.4.1.

Key words and phrases: Petri nets, First order logic, Reachability graph.

* A short version of this paper was published at FSTTCS 2011.

We provide a robust proof schema that entails undecidability of most logical fragments interpreted on such graphs.

Our motivations. For Petri nets, model checking CTL formulae with atomic propositions of the form $p > 0$ (place p contains at least one token) is known to be undecidable [13]. This negative result carries over to all fragments of CTL containing the modalities EF or AF. Furthermore, model checking CTL without atomic propositions but with next-time modalities indexed by action labels is undecidable too [13]. In contrast, LTL model-checking over vector addition systems with states is EXPSpace-complete [19] when atomic propositions refer to control states.

These negative results do not compromise the search for decidable fragments of first-order logic that describe, only purely graph-theoretically, the shape of the Petri net state graphs. So we intentionally avoid edge labels and atomic propositions interpreted on markings. As an example, we shall consider the first-order structure $(\mathbb{N}^n, \rightarrow)$ derived from a Petri net N with n places such that $M \rightarrow M'$ iff M evolves to M' by firing a transition of N . Since $(\mathbb{N}^n, \rightarrow)$ is an automatic structure, its first-order theory over predicates \rightarrow and $=$ is decidable, see e.g. [6]. This decision procedure can be extended to Petri net state graphs with Presburger-definable predicates on markings and with labels on transitions. As a second example of results related to our work, given a formula φ in $\text{FO}(\rightarrow, =)$ with free variables x_1, \dots, x_m , one can effectively construct a Presburger formula that characterizes exactly the markings satisfying φ in $(\mathbb{N}^n, \rightarrow)$.

However, it is unclear what happens if we consider the first-order theory of \rightarrow over the practically interesting structure $(\text{Reach}(N), \rightarrow)$. Here, $\text{Reach}(N)$ denotes the set of all markings *reachable* from the initial marking of Petri net N . Our paper studies this problem. We investigate the decidability status of several first-order logics, sometimes extended by a bit of MSO (via reachability predicates), sharing with [40] a common motivation. The properties of the reachability graph we are interested in are *purely graph-theoretical* in that they do not refer to tokens or transition labels and they are mostly *local* in that we often restrict ourselves to \rightarrow instead of its transitive closure. As summarised in Table 1 (Section 5) we settle the decidability status of most problems. To the best of our knowledge, this is the first study of logics for the reachability graph. In particular, related logics in [3] consider quantitative properties on markings and transitions, and evaluate formulae on runs. We do not refer to tokens or to transition labels.

Our contributions. We investigate the model-checking problem over structures of the form $(\text{Reach}(N), \rightarrow, \overset{*}{\rightarrow})$ generated from Petri nets N with first-order languages including predicate symbols for \rightarrow and/or $\overset{*}{\rightarrow}$. We consider variants depending on the predicates and on whether $\text{Reach}(N)$ or $\overset{*}{\rightarrow}$ are effectively semilinear. This allows us to provide a refined analysis about the decidability borders for such problems. As it is a classical fragment of first-order logic, we also consider the modal language $\text{ML}(\square, \square^{-1})$ with forward and backward modalities. Let us mention some features of our investigation:

- (1) Undecidability proofs are obtained by reduction from the equality problem (or the inclusion problem) between reachability sets defined by Petri nets, shown undecidable in [4, 21]. We demonstrate that our proof schema is robust and can be adapted to numerous formalisms specifying local properties as in first-order logic. Moreover, undecidability can be obtained even for a fixed formula (i.e., for a fixed property).

- (2) To determine the cause of undecidability, we investigate logical fragments. At the same time, we strive for maximally expressive decidable fragments. With these two goals, our study on graph-theoretical properties is quite systematic.
- (3) For decidable problems, we assess the computational complexity — either relative to standard complexity classes such as PSPACE or EXPSpace or by establishing a reduction from the reachability problem for Petri nets (when decision procedures rely on solving instances of this problem).

Our main findings are as follows (refined statements can be found in the body of the paper, see also Table 1 in Section 5):

- ★ Model-checking $(\text{Reach}(N), \rightarrow)$ [resp. $(\text{Reach}(N), \overset{*}{\rightarrow})$, $(\text{Reach}(N), \overset{\pm}{\rightarrow})$] is undecidable for the corresponding first-order language with a single binary predicate symbol.
- ★ Undecidability is also shown for the positive fragment of $\text{FO}(\rightarrow)$, for the forward fragment of $\text{FO}(\rightarrow)$, and for $\text{FO}(\rightarrow)$ augmented with $\overset{*}{\rightarrow}$. The latter result even holds if the reachability sets are effectively semilinear.
- ★ Combining procedures for coverability and reachability in Petri nets, we obtain some positive results. We prove that model-checking the existential fragment of $\text{FO}(\rightarrow)$ is decidable, but as hard as the reachability problem for Petri nets. Moreover, the model checking problem is decidable for $\text{FO}(\rightarrow, \overset{*}{\rightarrow}, =)$ under the assumption that the relations \rightarrow and $\overset{*}{\rightarrow}$ are semilinear (consequence of [6]). We have not found any decision result between these two extremes.
- ★ Concerning the modal language $\text{ML}(\Box, \Box^{-1})$, the global model-checking problem on $(\text{Reach}(N), \rightarrow)$ is undecidable but it becomes decidable when restricted to $\text{ML}(\Box)$ (even if extended with Presburger-definable predicates on markings); the latter problem is also as hard as the reachability problem for Petri nets.

One may regret that our main results turn towards undecidability but this was not clear at all when we began our study. On the positive side, we were able to identify non-trivial fragments for which the decision problems can be of high computational complexity. Our results shed some new light on the verification of structural properties on unlabelled net reachability graphs.

Structure of the paper. The remaining sections are organized as follows. Section 2 brings the background of the study. Section 3 presents results that focus on the reachability graph without the reachability predicate. Section 4 presents those involving the reachability predicate.

2. PRELIMINARIES

We recall basics on Petri nets and semilinear sets and we give the standard definitions and fundamental results used in the paper. We first introduce the notations needed when considering Petri net reachability graphs as models for first-order sentences. Then, we define first-order logic and modal logic interpreted on graphs induced by Petri nets. Finally, we present positive decidability results about model-checking problems.

2.1. Petri nets. A *Petri net* is a bi-partite graph $N = (P, T, F, M_0)$, where P and T are finite disjoint sets of *places* and *transitions*, and $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a set of directed edges with non-negative integer weights. A *marking* of N is a function $M : P \rightarrow \mathbb{N}$. M_0 is the *initial marking* of N . A transition $t \in T$ is *enabled* at a marking M , written $M[t]$, if $M(p) \geq F(p, t)$ for all places $p \in P$. If t is enabled at M then it can be *fired*. This leads to the marking M' defined by $M'(p) = M(p) + F(t, p) - F(p, t)$ for all $p \in P$. The firing relation is denoted by $M[t]M'$. The definitions are extended to transition sequences $s \in T^*$ in the expected way. A marking M' is *reachable* from a marking M if $M[s]M'$ for some $s \in T^*$. A transition t is *in self-loop* with a place p iff $F(p, t) = F(t, p) > 0$. A transition is *neutral* if it has null effect on all places. The *reachability set* $\text{Reach}(N)$ of N is the set of all markings that are reachable from the initial marking.

Theorem 2.1. [35] *Given a Petri net N and two markings M and M' , one can decide whether M' is reachable from M .*

Theorem 2.2. [4, 21] *Given two Petri nets N and N' , it is undecidable whether $\text{Reach}(N) = \text{Reach}(N')$ [resp. $\text{Reach}(N) \subseteq \text{Reach}(N')$].*

A stronger version of Theorem 2.2 has been established in [28] where it was shown that undecidability still holds when N and N' have five places and one of these nets is fixed.

A Petri net $N = (P, T, F, M_0)$ induces several standard structures on which first-order logics may be interpreted. The *plain unlabelled reachability graph* of N is the structure $\text{PURG}(N) = (D, \rightarrow)$ where $D = \text{Reach}(N)$ and \rightarrow is the binary relation on D defined by $M \rightarrow M'$ if $M[t]M'$ for some $t \in T$. Note that $M_0 \in D$ but no predicate is given to identify this specific marking. The *unlabelled reachability graph* of N is the structure $\text{URG}(N) = (D, \text{init}, \rightarrow, \overset{*}{\rightarrow}, \overset{+}{\rightarrow}, =)$ where $\text{init} = \{M_0\}$, and relations $\overset{*}{\rightarrow}$ and $\overset{+}{\rightarrow}$ are the iterative and strictly iterative closures of \rightarrow , respectively. The *unlabelled transition graph* of N is the structure $\text{UG}(N) = (\mathbb{N}^P, \text{init}, \rightarrow, \overset{*}{\rightarrow}, \overset{+}{\rightarrow}, =)$ where $M \rightarrow M'$ if $M[t]M'$ for some transition $t \in T$. Note that reachability of markings is not taken into account in $\text{UG}(N)$. In the sequel, by default $\text{card}(P) = n$ and we identify \mathbb{N}^P and \mathbb{N}^n . We also call *1-loop* an edge $M \rightarrow M'$ with $M = M'$.

2.2. Petri nets and semilinear sets. We rely on results about the semilinear subsets of \mathbb{N}^n that represent possible markings of a Petri net with n places. Recall that $(\mathbb{N}^n, +)$ is a commutative monoid where the product operation is the componentwise addition of n -vectors $(+)$ and the neutral element is the null n -vector.

A subset $E \subseteq \mathbb{N}^n$ is called *linear* if it can be expressed as $x + \{y_1, \dots, y_m\}^*$ for vectors $x \in \mathbb{N}^n$ and $y_1, \dots, y_m \in \mathbb{N}^n$. The Kleene iteration $\{y_1, \dots, y_m\}^*$ is a shorthand notation for $k_1 y_1 + \dots + k_m y_m$ for some $k_1, \dots, k_m \in \mathbb{N}$. A subset $E \subseteq \mathbb{N}^n$ is *semilinear* if it is a finite union of linear subsets. Owing to the commutativity of the product operation $+$, semilinear subsets of \mathbb{N}^n coincide with the regular subsets of \mathbb{N}^n . Hence, they are generated by finite automata over \mathbb{N}^n . Indeed, one can always choose finite automata whose transitions are labelled with generators, i.e., with n -vectors with a single non-null entry equal to 1. The semilinear subsets of \mathbb{N}^n form an effective Boolean algebra [16], hence providing decision procedures for emptiness. In [17], Ginsburg and Spanier gave an effective correspondence between semilinear subsets and Presburger subsets, i.e., subsets of \mathbb{N}^n definable in Presburger arithmetic. Presburger arithmetic can be decided in triple exponential time [8].

Proposition 2.3. *Given a Petri net $N = (P, T, F, M_0)$ and a semilinear subset of markings $E \subseteq \mathbb{N}^{|P|}$, one can decide whether (some marking in) E can be reached from M_0 .*

Hack reduced this semilinear reachability problem to the reachability problem in Petri nets [21, Lemma 4.3]. The proposition now follows with the decidability of reachability in Theorem 2.1. The statement shows in particular that for any marking $M \in \mathbb{N}^{|P|}$, one can decide whether a marking greater than or equal to M is reachable.

We recalled in the introduction that it is decidable whether the reachability set of a Petri net system is semilinear. Note that semilinearity of the reachability set $\text{Reach}(N)$ does not entail semilinearity of the reachability relation $\xrightarrow{*} \subseteq \text{Reach}(N) \times \text{Reach}(N) \subseteq \mathbb{N}^{|P|+|P|}$. Here are some classes of Petri nets and counter systems for which the reachability relation $\xrightarrow{*}$ is effectively semilinear (apart from bounded Petri nets):

- ★ Cyclic Petri nets, see e.g. [2, 9, 32].
- ★ Communication-free Petri nets [12].
- ★ Vector addition systems with states of dimension 2 [25, 33].
- ★ Single-path Petri nets [26].
- ★ Petri nets with regular languages [41].
- ★ Flat affine counter systems with the finite monoid property [7, 14].
- ★ Flat relational counter systems [11, 10].
- ★ Reversal-bounded counter systems [27].

Some of these results require complex machinery but they are essential to use the decision procedures based on effective semilinearity.

2.3. First-order languages. To specify properties of structures $\text{URG}(N)$, $\text{PURG}(N)$ and $\text{UG}(N)$ obtained from a Petri net N , we introduce a first-order logic FO with atomic predicates $x \rightarrow y$, $x \xrightarrow{*} y$, $x \xrightarrow{+} y$ and $\text{init}(x)$. Formulae in FO are defined by

$$x \rightarrow y \mid x \xrightarrow{*} y \mid x \xrightarrow{+} y \mid \text{init}(x) \mid x = y \mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x \varphi \mid \forall x \varphi.$$

Given a set P of predicate symbols from the above signature, we denote the *restriction of FO to the predicates in P* by $\text{FO}(P)$. By default, FO refers to the full language. Formulae are interpreted either on $\text{PURG}(N)$, $\text{URG}(N)$ or $\text{UG}(N)$. Observe that FO on $\text{UG}(N)$ enables, using init and reachability predicates, to relativize formulae to $\text{URG}(N)$, but restricted logical languages motivate the existence of both structures. It is worth noting that by slight abuse, we sometimes use the same notation for a predicate symbol and its fixed interpretation. Note that, as regards interpretation, $\xrightarrow{*} = (= \cup \xrightarrow{+})$ and $\xrightarrow{+} = (\rightarrow \circ \xrightarrow{*})$, hence $\text{FO}(\text{init}, \rightarrow, \xrightarrow{+}, =)$, $\text{FO}(\text{init}, \rightarrow, \xrightarrow{*}, =)$, and $\text{FO}(\text{init}, \rightarrow, \xrightarrow{+}, \xrightarrow{*}, =)$ are equally expressive. FO indicates that one can quantify over markings. Note that predicates $\xrightarrow{+}$ or $\xrightarrow{*}$ exceed the expressiveness of usual first-order logics on graphs. We omit the standard definition of the satisfaction relation $\mathcal{U}, \mathbf{v} \models \varphi$ with \mathcal{U} a structure ($\text{PURG}(N)$, $\text{URG}(N)$ or $\text{UG}(N)$) and \mathbf{v} a valuation of the free variables in φ . For example, $\forall x \varphi$ holds true whenever the formula φ holds true for all elements (markings) of the considered structure. *Sentences* are closed formulae, i.e., without free variables. If $\mathcal{U} \models \varphi$ then \mathcal{U} is called a model of φ .

It is worth noting that FO can only describe *graph-theoretical* properties of the structures \mathcal{U} , apart from equality tests. The binary relations do not use transitions of nets as labels and no atomic propositions give reference to markings. As a consequence, quantitative properties about markings cannot be expressed in FO, at least in the obvious way, and

constraints about the firing of specific transitions cannot be expressed either. Note that FO is not minimal when it comes to expressiveness. The redundancies, however, help us design interesting logical fragments.

In the sequel, we consider several model-checking problems. The model-checking problem $\text{MC}^{\text{URG}}(\text{FO})$ is stated as follows:

input:: a Petri net $N = (P, T, F, M_0)$ and a sentence $\varphi \in \text{FO}$
question:: $\text{URG}(N) \models \varphi?$

The variant $\text{MC}^{\text{UG}}(\text{FO})$ is:

input:: a Petri net $N = (P, T, F, M_0)$ and a sentence $\varphi \in \text{FO}$
question:: $\text{UG}(N) \models \varphi?$

The logics $\text{FO}(\mathbf{P})$ (atomic formulae restricted to predicates in \mathbf{P}) induce restricted variants of the two model checking problems that we denote by $\text{MC}^{\text{URG}}(\text{FO}(\mathbf{P}))$ and $\text{MC}^{\text{UG}}(\text{FO}(\mathbf{P}))$, respectively. Formulae in FO can express standard structural properties, for instance deadlock-freeness with $\forall x \exists y x \rightarrow y$, existence of a 1-loop with $\exists x x \rightarrow x$, or cyclicity with $\forall x \forall y x \xrightarrow{*} y \Rightarrow y \xrightarrow{*} x$. Automatic structures form a large class of structures having a decidable model checking problem for FO. These structures have presentations in which k -ary relations are defined by synchronous automata (see [6] for more details).

Theorem 2.4. [6] *Let \mathcal{S} be an automatic structure, then $\text{MC}^{\mathcal{S}}(\text{FO})$ is decidable.*

From [16], semilinear sets and semilinear relations are automatic. In particular, this means that $(\mathbb{N}^n, \rightarrow, =)$ is automatic. Propositions 2.5, 2.6 and 2.7 are consequences of Theorem 2.4; they are provided below to present more explicitly what is the current state of knowledge.

Proposition 2.5. $\text{MC}^{\text{UG}}(\text{FO}(\rightarrow, =))$ is decidable.

Note that given φ in $\text{FO}(\rightarrow, =)$, one can effectively build a Presburger formula that characterizes exactly the valuations satisfying φ in $\text{UG}(N)$. Decidability is preserved with Presburger-definable properties on markings and with labelled transition relations $\{t\}$. However, having \mathbb{N}^n as a domain does not always guarantee decidability, see the undecidability result in [40, Theorem 2] about a structure with domain \mathbb{N}^n but equipped with successor relations for each dimension and with reachability predicates constrained by regular languages. Likewise, subproblems of $\text{MC}^{\text{URG}}(\text{FO})$ may require additional assumptions to achieve decidability, as the semilinearity assumption made in the statement below. The proposition also follows from Theorem 2.4.

Proposition 2.6. *Let \mathcal{C} be a class of Petri nets for which the restriction on reachable markings of the reachability relation $x \xrightarrow{*} y$ is effectively semilinear. Then, $\text{MC}^{\text{URG}}(\text{FO})$ restricted to \mathcal{C} is decidable.*

Proof. Let $N = (P, T, F, M_0)$ be a Petri net in \mathcal{C} with $\text{card}(P) = n$. We represent its markings by vectors $M \in \mathbb{N}^n$. By assumption, $\text{Reach}(N)$ and the set $\{(M, M') \mid M, M' \in \text{Reach}(N) \text{ and } M \xrightarrow{*} M'\}$ are effectively semilinear. Similarly, the set $\{(M, M) \mid M \in \text{Reach}(N)\}$ is effectively semilinear. Define $\Delta = \{(M, M') \mid M, M' \in \text{Reach}(N) \text{ and } M \xrightarrow{*} M', M \neq M'\}$. Then Δ is effectively semilinear. Let $\Delta^2 = \{(M, M') \mid (\exists M'') (M, M'' \in \Delta \text{ and } (M'', M') \in \Delta)\}$. As semilinear sets are closed under projection (quantifier elimination in Presburger arithmetic), Δ^2 is effectively semilinear. Now $\{(M, M') \mid M \in \text{Reach}(N) \text{ and}$

$M \xrightarrow{\pm} M'$ is equal to $\Delta \cup \Delta^2$. Hence this set is effectively semilinear. Therefore, through the effective correspondence between semilinear sets and sets definable in Presburger arithmetic, any sentence φ of FO translates to a sentence φ' of Presburger arithmetic logic such that $\text{URG}(N) \models \varphi$ if and only if φ' is true. The proposition follows from the decidability of Presburger arithmetic [39]. \square

When reachability sets are effectively semilinear but the reachability relation is not, the strictly less expressive logical fragment $\text{FO}(\rightarrow, =)$ remains decidable, from Theorem 2.4.

Proposition 2.7. *Let \mathcal{C} be a class of Petri nets N for which $\text{Reach}(N)$ is effectively semilinear. Then, $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow, =))$ restricted to \mathcal{C} is decidable.*

Proof. Consider a Petri net $N = (P, T, F, M_0)$ in \mathcal{C} . Assume the Presburger formula $\varphi(x_1, \dots, x_n)$ characterizes $\text{Reach}(N)$ where $|P| = n$. There is a second Presburger formula $\varphi'(x_1, \dots, x_n, x'_1, \dots, x'_n)$ that characterizes the binary relation \rightarrow in $\text{UG}(N)$.

Given a sentence ψ in $\text{FO}(\rightarrow, =)$, one can build a sentence $f(\psi)$ in Presburger arithmetic such that $\text{URG}(N) \models \psi$ iff $f(\psi)$ is satisfiable in Presburger arithmetic. The map $f(\cdot)$ is homomorphic for Boolean connectives. Furthermore,

$$\begin{aligned} \star f(z \rightarrow z') &\stackrel{\text{def}}{=} \varphi'(z_1, \dots, z_n, z'_1, \dots, z'_n), \\ \star f(z = z') &\stackrel{\text{def}}{=} \bigwedge_{i \in [1, n]} z_i = z'_i, \\ \star f(\forall z \chi) &\stackrel{\text{def}}{=} \forall z_1, \dots, z_n (\varphi(z_1, \dots, z_n) \Rightarrow f(\chi)). \end{aligned}$$

To evaluate predicate \rightarrow , we resort to φ' . With φ , we relativize the quantifiers to taking only positions in $\text{Reach}(N)$ into account. \square

Again, decidability is preserved with Presburger-definable properties on markings and with labelled transition relations of the form \xrightarrow{t} . To give an example application of this result, $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow, =))$ restricted to cyclic Petri nets is decidable. This follows from Proposition 2.7 combined with the fact that cyclic Petri nets have semilinear reachability sets [9]. The restriction to language $\text{FO}(\rightarrow, =)$ is essential for the decidability in Proposition 2.7. As we shall see in Proposition 4.5, the related model checking problem $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow, \overset{*}{\rightarrow}))$ is undecidable — even under the assumption of semilinearity for the reachability sets.

2.4. Standard first-order fragments: modal languages. By moving along edges, modal languages provide a local view to (potentially labelled) graph structures. Note the contrast to first-order logic in which one quantifies over any element of the structure. Applications of modal languages include modelling temporal and epistemic reasoning, and they are central for designing logical specification languages. In this paper, we consider simple modal languages understood as distinguished fragments of first-order logic. Moreover, the modal language ML defined below has no propositional variable (like Hennessy-Milner modal logic [23] but unlike standard modal logic K [5]) and no label on modal operators (unlike in modal languages dedicated to describing labelled transition systems). This allows us to interpret modal formulae on directed graphs of the form $(\text{Reach}(N), \rightarrow)$. However, in some places, we shall indicate when decidability or complexity results can be extended to richer versions of ML. The modal formulae in ML are defined by the grammar

$$\perp \mid \top \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box\varphi \mid \Diamond\varphi \mid \Box^{-1}\varphi \mid \Diamond^{-1}\varphi.$$

This language is not only poor compared to first-order logic, but also little expressive compared to other modal dialects. Yet, it is sometimes sufficiently expressive to obtain first undecidability results for model checking Petri net structures. Given a modal formula φ , its *modal degree* is the greatest number of nested occurrences of modal operators in φ . We write $\text{ML}(\square)$ to denote the restriction of ML to the modal operators \square and \diamond . We interpret modal formulae on directed graphs of the form (D, \rightarrow) for some Petri net $N = (P, T, F, M_0)$ with $\text{URG}(N) = (D, \text{init}, \rightarrow, \xrightarrow{*}, \xrightarrow{+}, =)$. We provide the definition of the satisfaction relation \models relatively to an arbitrary directed graph $\mathcal{M} = (W, R)$ (and $w \in W$). The clauses for Boolean connectives and logical constants are standard and we omit them. For the modal operators, we set

- * $\mathcal{M}, w \models \square\varphi \stackrel{\text{def}}{\iff}$ for every $w' \in W$ such that $(w, w') \in R$, we have $\mathcal{M}, w' \models \varphi$.
- * $\mathcal{M}, w \models \diamond\varphi \stackrel{\text{def}}{\iff}$ there is $w' \in W$ such that $(w, w') \in R$ and $\mathcal{M}, w' \models \varphi$.
- * $\mathcal{M}, w \models \square^{-1}\varphi \stackrel{\text{def}}{\iff}$ for every $w' \in W$ such that $(w', w) \in R$, we have $\mathcal{M}, w' \models \varphi$.
- * $\mathcal{M}, w \models \diamond^{-1}\varphi \stackrel{\text{def}}{\iff}$ there is $w' \in W$ such that $(w', w) \in R$ and $\mathcal{M}, w' \models \varphi$.

As usual, \square and \diamond as well as \square^{-1} and \diamond^{-1} are dual operators that can be defined one from another as soon as negation is part of the language.

The model-checking problem $\text{MC}^{\text{URG}}(\text{ML})$ is the following:

- input::** a Petri net $N = (P, T, F, M_0)$ and a modal formula $\varphi \in \text{ML}$.
question:: $(\text{Reach}(N), \rightarrow), M_0 \models \varphi?$

Let $\text{MC}^{\text{URG}}(\text{ML}(\square))$ denote $\text{MC}^{\text{URG}}(\text{ML})$ restricted to $\text{ML}(\square)$. Proposition 2.8 proves this model checking problem decidable. The procedure exploits the fact that a modal formula of modal degree d can only induce constraints on nodes at distance at most d from the initial marking, a standard argument, see e.g. [5].

Proposition 2.8. $\text{MC}^{\text{URG}}(\text{ML}(\square))$ is decidable and PSPACE-complete.

Proof. Consider a Petri net $N = (P, T, F, M_0)$ with $\text{URG}(N) = (D, \text{init}, \rightarrow, \xrightarrow{*}, \xrightarrow{+}, =)$. Let φ be a modal formula in $\text{ML}(\square)$ with modal degree d (d is the greatest number of nested occurrences of modal operators in φ). We consider the directed graph $\mathcal{M} = (W, R)$ so that

- * $W \subseteq \mathbb{N}^P$ and R is the restriction of \rightarrow to W .
- * For $M \in \mathbb{N}^P$ we set $M \in W \stackrel{\text{def}}{\iff}$ there is a sequence of transitions s of length at most d such that $M_0[s]M$.

Observe that \mathcal{M} is finite and the cardinal of W is at most exponential in the size of N and d . One can show that $\mathcal{M}, M_0 \models \varphi$ iff $(D, \rightarrow), M_0 \models \varphi$. Hence, $\text{MC}^{\text{URG}}(\text{ML}(\square))$ is decidable, because the model-checking problem for ML over finite structures is decidable (in polynomial time). The PSPACE upper bound can be obtained with an algorithm similar to the one that shows CTL model-checking over 1-safe Petri nets to be in PSPACE, see e.g. [13, Section 4.2]. Our problem is actually simpler since we can restrict ourselves to the temporal operators **AX** and **EX** corresponding to \square and \diamond , respectively. We briefly describe below the nondeterministic algorithm $\text{MC}((P, T, F, M_0), \varphi)$ that returns **true** whenever $(D, \rightarrow), M_0 \models \varphi$. We proceed by a case analysis.

- $\varphi = \top$ **return true**;
 $\varphi = \neg\varphi'$: **if** $\text{MC}((P, T, F, M_0), \varphi')$ **then return false else return true**;

$\varphi = \varphi_1 \wedge \varphi_2$: **if** $MC((P, T, F, M_0), \varphi_1)$ **and** $MC((P, T, F, M_0), \varphi_2)$ **then return true else return false**;

$\varphi = \Box\varphi'$: **if** for some M' such that $M_0 \xrightarrow{t} M'$ with $t \in T$ we have $MC((P, T, F, M'), \varphi') = \text{false}$ **then return false else return true**.

Note that the depth of recursive calls for $MC((P, T, F, M_0), \varphi)$ is bounded by the modal degree of φ and each call requires only polynomial space in the size of (P, T, F, M_0) and φ . Hence, $MC((P, T, F, M_0), \varphi)$ runs in nondeterministic polynomial space. By Savitch Theorem, we get the bound PSPACE.

To establish PSPACE-hardness, we give a reduction from QBF. Let $\mathcal{Q}_1 p_1 \cdots \mathcal{Q}_{2n} p_{2n} \psi$ be a QBF formula where $\mathcal{Q}_1 \cdots \mathcal{Q}_{2n}$ is a sequence of quantifiers starting with $\mathcal{Q}_1 = \exists$, alternating strictly \exists and \forall , and ψ is a quantifier-free propositional formula built over the propositional variables in $\{p_1, \dots, p_{2n}\}$. We consider a modal formula φ of the form $(\Diamond\Box)^n \psi'$ where ψ' is obtained from ψ by replacing each propositional variable p_i by $\Diamond^i \Box \perp$. Construct a Petri net $N = (P, T, F, M_0)$ as follows. The set of places P contains a subset $\{p_1, \dots, p_{2n}\}$, in bijection with the atomic propositions and initially empty, plus auxiliary places. From M_0 , N executes first a sequence of $2n$ independent choices $(t'_1 + t''_1) \cdot (t'_2 + t''_2) \cdot \dots \cdot (t'_{2n} + t''_{2n})$ where t'_i puts i tokens in place p_i to represent the truth of the corresponding atomic proposition while t''_i puts no tokens in p_i to indicate the proposition does not hold. After this sequence of binary choices, N executes a non-deterministic choice $(x_1 + \dots + x_{2n})$ where x_i removes one token from p_i and puts one token in a place p'_i which was initially empty. Each control place p'_i is set in self-loop with a transition t_i that removes at each firing one token from p_i .

Existential quantifications are replaced by \Diamond , and universal ones by \Box . A path relative to a formula $(\Diamond\Box)^n$ then ends up in a configuration where truth values have been chosen for all variables. Note that the formula needs to be true for one continuation at each \Diamond position and true for each continuation at \Box positions. The last part of the formula needs to check the truth values of individual variables. For each p_i , we have a formula $\Diamond^i \Box \perp$ that is true only when there is precisely a path of length i , which corresponds to our encoding of truth values. The selection of each individual variable (and only one) is performed by the transition $(x_1 + \dots + x_{2n})$. Altogether, $(\text{Reach}(N), \rightarrow), M_0 \models (\Diamond\Box)^n \psi'$ iff $\mathcal{Q}_1 p_1 \cdots \mathcal{Q}_{2n} p_{2n} \psi$ is satisfiable. Note that $\text{Reach}(N)$ is finite. \square

For simple models (like finite structures), adding \Box^{-1} to $\text{ML}(\Box)$, often does not change the decidability status or the computational complexity of model checking, see e.g. [5]. When it comes to Petri net reachability graphs $\text{PURG}(N)$, adding the backward operator \Box^{-1} preserves decidability but at the cost of performing reachability checks.

Proposition 2.9. $\text{MC}^{\text{URG}}(\text{ML}(\Box, \Box^{-1}))$ is decidable.

Proof. Consider a Petri net $N = (P, T, F, M_0)$ with $\text{URG}(N) = (D, \text{init}, \rightarrow, \overset{*}{\rightarrow}, \overset{\perp}{\rightarrow}, =)$. Let φ be a modal formula in $\text{ML}(\Box, \Box^{-1})$ of modal degree d . Define $\overline{N} = (P, T \cup T^{-1}, F, M_0)$ where T^{-1} is a set of formal inverses of the transitions in T , i.e., $F(p, t^{-1}) = F(t, p)$ and $F(t^{-1}, p) = F(p, t)$ for all $t \in T$. To model check $\text{URG}(N)$ against φ , the idea is to consider a depth d unrolling of $\text{URG}(\overline{N})$. However, when following inverse transitions $M'[t^{-1}]M$, reachability checks are needed to guarantee the target marking M belongs to the domain D of structure $\text{URG}(N)$. These checks are effective by Theorem 2.1 quoted from [35, 30, 31]. More formally, we consider the directed graph $\mathcal{M}' = (W', R')$ defined by

- ★ $W' \subseteq \mathbb{N}^P$ and R' is the restriction of \rightarrow to W' .
- ★ For $M \in \mathbb{N}^P$ we set $M \in W' \stackrel{\text{def}}{\iff}$
 - (a) $M \in D$,
 - (b) there is a sequence of transitions $s \in (T \cup T^{-1})^*$ of length at most d such that $M_0[s]M$.

Checking $M_0[s]M$ is easy whereas $M \in D$ requires a reachability check. Observe that \mathcal{M}' is finite and effectively constructible. The cardinal of W' is exponential in d . One can show that $\mathcal{M}', M_0 \models \varphi$ iff $(D, \rightarrow), M_0 \models \varphi$. Hence, $\text{MC}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$ is decidable, because model-checking ML over finite structures is a decidable problem that takes polynomial time. \square

The best known decision procedures for Petri net reachability are non primitive recursive, which provides the worst possible and hopefully not tight upper bound to the complexity of the model-checking problem $\text{MC}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$. Unfortunately, it might well be the case that this upper complexity bound is tight, for we shall (in turn) reduce Petri net reachability to the above model-checking problem in Section 3.4.

We introduce another decision problem about ML that is closely related to first-order model-checking over reachability graphs. The *validity problem* $\text{VAL}^{\text{URG}}(\text{ML})$, also known as *global model-checking*, is stated as follows:

- input::** a Petri net $N = (P, T, F, M_0)$ that induces the structure $\text{URG}(N) = (D, \text{init}, \rightarrow, \overset{*}{\rightarrow}, \overset{+}{\rightarrow}, =)$, and a modal formula $\varphi \in \text{ML}$.
- question::** $(D, \rightarrow), M \models \varphi$ for every marking $M \in D$?

As observed earlier, formulae from $\text{ML}(\square, \square^{-1})$ can be viewed as first-order formulae in $\text{FO}(\rightarrow)$. Therefore, using modal languages in specifications is a way to consider fragments of $\text{FO}(\rightarrow)$. Indeed, given a modal formula φ in $\text{ML}(\square, \square^{-1})$, one can compute in linear time a first-order formula φ' with only two individual variables (see e.g. [5]) that satisfies: for every Petri net N we have $\text{PURG}(N) \models \varphi'$ iff $\text{PURG}(N), M \models \varphi$ for every marking M in $\text{Reach}(N)$. Hence, the validity problem $\text{VAL}^{\text{URG}}(\text{ML})$ appears as a natural counterpart to the model-checking problem for FO over unlabelled reachability graphs of Petri nets. We will see in the next section that both problems are undecidable.

We conclude the section by introducing an extension of ML that admits quantifier-free formulae from Presburger arithmetic as atomic propositions. The idea is to pose arithmetical constraints on the numbers of tokens in places, and thus to increase the expressiveness of ML. We call this logic PAML and it will be mainly used in decidability results in Section 3.3. The domain of the structure for PAML needs to be of the form \mathbb{N}^P . More precisely, with terms $t ::= a \times p \mid t + t$ where p is a place and $a \in \mathbb{Z}$ we define PAML from ML by adding atomic formulae ψ defined by

$$\psi ::= \top \mid t \leq k \mid t \geq k \mid t \equiv_c k' \mid \psi \wedge \psi \mid \neg\psi.$$

Here, \top is the truth constant, $c \in \mathbb{N} \setminus \{0, 1\}$, $k \in \mathbb{Z}$ and $k' \in \mathbb{N}$. The definition of $(\text{Reach}(N), M) \models \psi$ depends on the definition of satisfaction of ψ in Presburger arithmetic by a tuple M . The details are as expected and we omit them here. It can be shown that $\text{MC}^{\text{URG}}(\text{PAML}(\square, \square^{-1}))$ is decidable. The proof is similar to the proof of Proposition 2.9.

3. STRUCTURAL PROPERTIES OF UNLABELLED NET REACHABILITY GRAPHS

We study the decidability status of model checking unlabelled reachability graphs of Petri nets against the first-order and modal logics defined in the previous section. Recall that the logics are designed to expressing purely graph-theoretical properties of reachability graphs.

3.1. A proof schema for undecidability of $\text{FO}(\rightarrow)$. To establish undecidability of $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$, model checking reachability graphs against first-order specifications, we provide a reduction of the equality problem for reachability sets. For two Petri nets N_1 and N_2 with identical sets of places, Hack proved it to be undecidable whether the sets of reachable markings $\text{Reach}(N_1)$ and $\text{Reach}(N_2)$ coincide (Theorem 2.2 recalls this result from [21]). To encode the equality problem into a first-order model checking problem, we join N_1 and N_2 in a third Petri net \overline{N} . The construction ensures that equality of the reachability sets can be checked with a first-order query: $\text{Reach}(N_1) = \text{Reach}(N_2)$ if and only if $\text{PURG}(\overline{N}) \models \varphi$. Interestingly, φ is a fixed formula and thus independent of the inputs N_1 and N_2 . Before we turn to the technicalities, we sketch the idea of the construction and comment on why it yields so much expressiveness. With an initial guess, \overline{N} decides to simulate either N_1 or N_2 . At any time, \overline{N} may stop the simulation. Then \overline{N} either starts behaving in different ways according to the initial choice between N_1 and N_2 . Alternatively, \overline{N} may forget this choice and enter a deadlock marking M that reflects the last marking of N_1 or N_2 in the simulation.

The reachability sets of N_1 and N_2 are equal if and only if every simulation result M can be obtained from both, N_1 and N_2 . But inspecting M in isolation does not reveal whether it stemmed from N_1 or N_2 . The idea is in the different behaviours that recall the initial guess when the simulation ends. They yield a neighbourhood of M in the reachability graph of \overline{N} that reveals the origin of the marking. Indeed, with *finite experiments* we can check whether M is found in the simulation of N_1 or N_2 . Equality of the reachability sets is then checked by a formula φ which requires that, for any simulation result M , both experiments witnessing for N_1 and N_2 succeed. The experiments consist of one backward transition and some forward transitions. Backward transitions reconstruct the initial choice, and forward transitions distinguish the nets N_1 and N_2 .

The strength of this construction stems from the combination of two ideas. A Petri net can (i) store choices over arbitrarily long histories and (ii) reveal this propagated information in local structures. These structures can be characterised by finite back and forth experiments that are expressed in terms of first-order formulae.

Construction. The two nets N_1 and N_2 to be compared for equality of reachability sets share all places. The constructed net, \overline{N} , has these places together with an initialization place p , two control places p_1 and p_2 , and additional places p'_1, p''_1 , and p'_2 that we will elaborate on below. The initialization place is the only place that is initially marked, by a single token.

As transitions, \overline{N} has the disjoint union of the transitions of N_1 and N_2 , plus additional transitions that we introduce now together with an explanation of their intended behaviour. The original transitions are put in self-loop with the respective control places. Furthermore, we have two concurrent transitions t_c^1, t_c^2 that consume the initial token and mark either p_1 and all places marked in the initial configuration of N_1 or p_2 and all places marked in the initial configuration of N_2 . Firing t_c^1 starts the simulation of N_1 , and similar for t_c^2 . Each subnet N_1 and N_2 may be stopped at any time by firing transitions t_{end}^1 and t_{end}^2 that move

the token from the control place p_1 or p_2 to the place p'_1 or p'_2 , respectively. As a result, the token count on the places of N_1 and N_2 is not changed any more.

When the transitions t_{end}^1 and t_{end}^2 have been fired, \overline{N} behaves as indicated in Figure 3.1 below M_1 and M_2 , respectively. At a marking M_1 , place p'_1 enables a transition t_ℓ^1 which puts a token on p''_1 , depicted by M_\odot in the figure. The place enables a transition t_{sl} in self-loop. Furthermore, two transitions t_{dl}^1 and t_{dl}^2 (from M_1 to M_ℓ and from M_2 to M_r) empty the places p'_1 and p'_2 . The markings reached by these transitions are designed to be deadlocks. Moreover, by construction of \overline{N} , deadlock markings can only be reached this way (as M_ℓ or M_r or both). Since, firing t_{dl}^1 or t_{dl}^2 lets \overline{N} forget the index 1 or 2 of the net that was simulated, we have the following relationship. Whenever a marking M is reached both in N_1 and N_2 , the corresponding markings in \overline{N} lead to $M_\ell = M_r$

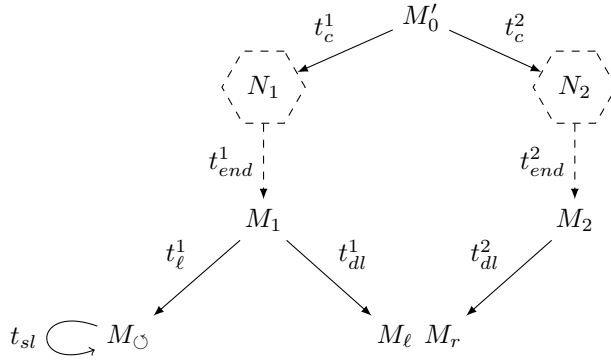


Figure 3.1: Reachability graph of \overline{N}

A formula expressing equality of the reachability sets of N_1 and N_2 (without recycling variables) is defined hereafter:

$$\varphi \stackrel{\text{def}}{=} \forall z (\neg \exists z' z \rightarrow z') \Rightarrow (\exists z_1 z_1 \rightarrow z \wedge \varphi_l(z_1)) \wedge (\exists z_2 z_2 \rightarrow z \wedge \neg \varphi_l(z_2))$$

Formula $\varphi_l(x) \stackrel{\text{def}}{=} \exists y (x \rightarrow y \wedge y \rightarrow x)$ indicates that x has a successor that has a 1-loop.

Lemma 3.1. $\text{Reach}(N_1) = \text{Reach}(N_2)$ if and only if $\text{PURG}(\overline{N}) \models \varphi$.

Proof. For the implication from left to right, consider a deadlock M . Marking M is reachable only via t_{dl}^1 or t_{dl}^2 , say $M_1[t_{dl}^1]M$. Then marking M_1 satisfies φ_l and stems from a marking $M'_1[t_{end}^1]M_1$ of N_1 . The hypothesis on equal reachability sets yields a marking M'_2 of N_2 that leads by transition t_{end}^2 to a marking M_2 satisfying $\neg \varphi_l$ as required.

In turn, if φ holds we establish two inclusions. To show $\text{Reach}(N_1) \subseteq \text{Reach}(N_2)$, consider marking M'_1 reachable via sequence s_1 in N_1 . In \overline{N} , the marking can be prolonged to a deadlock M with $M'_0[t_c^1]M'_0[s_1]M'_1[t_{end}^1]M_1[t_{dl}^1]M$. Here, M_1 satisfies φ_l . But φ yields another predecessor M_2 of M with $M_2 \neq M_1$. To avoid the 1-loop, marking M_2 has to result from a sequence $M'_0[t_c^2]M'_0[s_2]M'_2[t_{end}^2]M_2[t_{dl}^2]M$. It is readily checked that M'_1 and M'_2 coincide up to the token on the control place. Hence, $M'_1 \in \text{Reach}(N_2)$ as required. \square

Corollary 3.2. $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ is undecidable, already for the fixed formula φ given in this section.

By recycling variables in φ above, we get a sharp result that marks the undecidability border of model checking against $\text{FO}(\rightarrow)$ by two variables. Model checking $\text{FO}(\rightarrow)$ restricted to one variable is decidable.

Theorem 3.3. *There exists a formula φ in $\text{FO}(\rightarrow)$ with two individual variables such that $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ restricted to φ is undecidable.*

Proof. It is sufficient to observe that formula φ below

$$\forall z (\neg \exists z' z \rightarrow z') \Rightarrow (\exists z_1 z_1 \rightarrow z \wedge \varphi_l(z_1)) \wedge (\exists z_2 z_2 \rightarrow z \wedge \neg \varphi_l(z_2))$$

with $\varphi_l(x) \stackrel{\text{def}}{=} \exists y (x \rightarrow y \wedge y \rightarrow x)$ is logically equivalent to the formula

$$\forall z (\neg \exists z' z \rightarrow z') \Rightarrow (\exists z' z' \rightarrow z \wedge \varphi'_l(z')) \wedge (\exists z' z' \rightarrow z \wedge \neg \varphi'_l(z'))$$

where $\varphi'_l(z') \stackrel{\text{def}}{=} \exists z (z' \rightarrow z \wedge z \rightarrow z')$. Recycling of variables is explained e.g. in [15]. \square

Moreover, combined with the fact that model checking first order logic for automatic structures is decidable, Theorem 3.3 leads to the following impossibility result.

Corollary 3.4. *There is no algorithm to construct an automatic graph isomorphic to the unlabelled reachability graph of a Petri net.*

Note that this negative result cannot follow directly from complexity-theoretic considerations. Indeed, even if the unlabelled reachability graph of a Petri net could be represented as an automatic graph, this automatic graph could not be used to decide on reachability of markings unless this representation were in effective bijection with \mathbb{N}^n (where n is the number of places).

Restricted to a single variable, model checking $\text{FO}(\rightarrow)$ becomes decidable.

Proposition 3.5. $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ restricted to one individual variable is decidable.

Proof. Every sentence in $\text{FO}(\rightarrow)$ restricted to one individual variable is logically equivalent either to \perp , or to \top , or to a positive Boolean formula with atomic formulae of one of the forms below:

- (1) $\exists x (x \rightarrow x)$
- (2) $\exists x \neg(x \rightarrow x)$
- (3) $\forall x (x \rightarrow x)$
- (4) $\forall x \neg(x \rightarrow x)$.

Since (2) is the negation of (3) and (1) is the negation of (4), decidability is obtained by evaluating (1) $\text{PURG}(N) \models \exists x (x \rightarrow x)$ and (3) $\text{PURG}(N) \models \forall x (x \rightarrow x)$. (1) can be checked by solving one instance of the covering problem for each neutral transition of the net whereas (3) can be checked by solving a single instance of the reachability problem. Indeed, let T_\circ be the subset of transitions of the net that leave markings unchanged (neutral transitions). Then the set of markings specified hereafter is effectively semilinear:

$$Z \stackrel{\text{def}}{=} \{M : \text{not } M[t] \text{ for all } t \in T_\circ\}$$

We have $\text{not } \text{PURG}(N) \models \forall x (x \rightarrow x)$ iff there is a marking $M \in Z$ that is reachable, $M_0 \xrightarrow{*} M$. With [21, Lemma 4.3] this reduces to an instance of the reachability problem. \square

It is possible to play further with parameters. For instance, our undecidability proof uses several reachability graphs with constant formulae. It is open whether there is a fixed Petri net reachability graph for which the model-checking problem for $\text{FO}(\rightarrow)$ is undecidable.

3.2. Robustness of the proof schema. Based on the previous proof schema, this section presents undecidability results for subproblems of $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$. More specifically, we consider the positive fragment, the forward fragment, the restriction when the direction of edges is omitted, and $\text{ML}(\square, \square^{-1})$. For all these fragments, we establish undecidability of model checking.

3.2.1. Forgetting orientation. Let $\lambda(x, x') \stackrel{\text{def}}{=} (x \rightarrow x') \vee (x' \rightarrow x)$. Expressing properties about $\text{PURG}(N)$ in $\text{FO}(\lambda)$ amounts to getting rid of the direction of edges of this graph. Despite this weakening, undecidability is still present for general Petri nets. To instantiate the above argumentation, we have to identify deadlock markings and analyse their environment. In $\text{FO}(\lambda)$, we augment markings encountered during the simulation by 3-cycles. Then, the absence of 3-cycles and an environment without such cycles characterises deadlock markings.

Proposition 3.6. $\text{MC}^{\text{URG}}(\text{FO}(\lambda))$ is undecidable.

Proof. We take advantage of the fact that $\text{FO}(\lambda)$ can express that a node x belongs to an undirected cycle of length three. A possible formula is:

$$3\text{cycle}(x) \stackrel{\text{def}}{=} \exists y \exists z (\lambda(x, y) \wedge \lambda(y, z) \wedge \lambda(z, x)) \wedge \neg(\lambda(x, x) \vee \lambda(y, y) \vee \lambda(z, z))$$

Now consider two Petri nets N_1 and N_2 with identical sets of places. For $1 \leq i \leq 3$, add to each net new places p_i and transitions t_i such that p_1 contains initially one token, while p_2 and p_3 are empty. Transition t_i takes one token from p_i and puts one token in $p_{i+1 \bmod 3}$. The resulting Petri nets have identical reachability sets if and only if N_1 and N_2 have identical reachability sets. Therefore, equality of reachability sets is undecidable for nets in which every reachable marking belongs to some cycle of length three. Assuming that N_1 and N_2 have this property, let \overline{N} be the net constructed from N_1 and N_2 as in the proof of Proposition 3.3 (see also Figure 3.1). We can assume without loss of generality that every transition of N_1 and N_2 changes the current marking (the other transitions do not affect the reachability sets and can be removed). As a consequence, the reachability graphs of the augmented nets N_1 and N_2 have no 1-loops, which is required for the effectiveness of $3\text{cycle}(x)$. The deadlock markings of \overline{N} are then exactly the markings that have no cycle of length one or three and that are surrounded by nodes without cycles of length three:

$$\text{dead}(z) \stackrel{\text{def}}{=} \neg\lambda(z, z) \wedge \neg 3\text{cycle}(z) \wedge \forall x \lambda(z, x) \Rightarrow \neg 3\text{cycle}(x).$$

Equality of the reachability sets of N_1 and N_2 is then expressed by the formula φ below

$$\forall z \text{dead}(z) \Rightarrow (\exists z_1 \lambda(z, z_1) \wedge \varphi_1(z_1)) \wedge (\exists z_2 \lambda(z, z_2) \wedge \neg\varphi_1(z_2))$$

where $\varphi_1(z) \stackrel{\text{def}}{=} \exists y \lambda(z, y) \wedge \lambda(y, z)$. We have $\text{Reach}(N_1) = \text{Reach}(N_2)$ iff $\overline{N} \models \varphi$. By Theorem 2.2, $\text{MC}^{\text{URG}}(\text{FO}(\lambda))$ is undecidable. \square

3.2.2. *A well-known first-order fragment: $\text{ML}(\Box, \Box^{-1})$.* To establish undecidability of the problem $\text{VAL}^{\text{URG}}(\text{ML}(\Box, \Box^{-1}))$, we again provide a reduction from the equality problem for Petri net reachability sets.

Proposition 3.7. $\text{VAL}^{\text{URG}}(\text{ML}(\Box, \Box^{-1}))$ is undecidable.

Proof. Consider two Petri nets N_1 and N_2 with identical sets of places. We rely on the construction of \overline{N} in Section 3.1, but give a modal formula φ (independent of N_1 and N_2) that yields the following equivalence: N_1 and N_2 have the same reachability set iff $\text{PURG}(\overline{N}), M \models \varphi$ for every marking M in $\text{Reach}(\overline{N})$. For all deadlocks, there is one predecessor (from N_1) that is able to do two more steps and another predecessor (from N_2) that is not: $\varphi \stackrel{\text{def}}{=} \Box \perp \Rightarrow (\Diamond^{-1} \Diamond \Diamond \top \wedge \Diamond^{-1} \Box \Box \perp)$. Formula φ is semantically equivalent to the first-order formula φ_{fo} defined below:

$$\forall z (\neg \exists z' z \rightarrow z') \Rightarrow (\exists z_1, z_2, z_3 (z_1 \rightarrow z) \wedge (z_1 \rightarrow z_2) \wedge (z_2 \rightarrow z_3)) \wedge (\exists z_1 (z_1 \rightarrow z) \wedge \forall z_2, z_3 \neg((z_1 \rightarrow z_2) \wedge (z_2 \rightarrow z_3))). \quad \square$$

This undecidability result is tight. In Section 3.3.2, we establish decidability of an extended variant of $\text{VAL}^{\text{URG}}(\text{ML}(\Box))$ where the backward modality \Box^{-1} is excluded. Moreover, by translating formulae in $\text{ML}(\Box, \Box^{-1})$ to $\text{FO}(\rightarrow)$ restricted to two individual variables, we get another evidence that $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ restricted to two individual variables is undecidable.

3.2.3. *$\text{FO}(\rightarrow)$ restricted to positive or forward formulae.* Although $\text{VAL}^{\text{URG}}(\text{ML}(\Box, \Box^{-1}))$ and $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ are undecidable in general, we have identified decidable fragments of modal logic in Section 2.4. By analogy, one may expect to find decidability of related fragments of first-order logic. We prove here that this is not the case. We consider forward $\text{FO}(\rightarrow)$ and positive $\text{FO}(\rightarrow)$ and show that their model checking problems are undecidable. In a *positive formula*, atomic propositions occur only under the scope of an even number of negations. Let $\text{FO}^+(\mathcal{P})$ denote the set of positive first order formulae over predicates in \mathcal{P} .

Proposition 3.8. $\text{MC}^{\text{URG}}(\text{FO}^+(\rightarrow))$ is undecidable.

Proof. We rely on the previously introduced proof schema. Let N_1 and N_2 be two Petri nets and \overline{N} their combination sketched in Figure 3.1. We propose a positive formula φ so that inclusion $\text{Reach}(N_2) \subseteq \text{Reach}(N_1)$ holds if and only if $\text{PURG}(\overline{N}) \models \varphi$:

$$\varphi \stackrel{\text{def}}{=} \forall z \exists z_1 \exists y_\ell \exists z' (z \rightarrow z') \vee ((z_1 \rightarrow z) \wedge (z_1 \rightarrow y_\ell) \wedge (y_\ell \rightarrow y_\ell))$$

The formula considers an arbitrary marking M . If M is no deadlock, nothing is required by φ . If M is a deadlock, then φ asks for vertices M_1 and M_\circ so that M_1 is a common direct ancestor of M and M_\circ and moreover M_\circ has a 1-loop.

By construction of \overline{N} , formula φ is satisfied if and only if every deadlock marking M reachable in \overline{N} (in particular, a simulation of N_2) can be reached in N_1 . This means $\text{Reach}(N_2) \subseteq \text{Reach}(N_1)$. \square

Open problem 1. Decidability status of $\text{MC}^{\text{URG}}(\text{FO}^+(\overset{*}{\rightarrow}))$. ○

A *forward formula* is a formula in which every occurrence $x \rightarrow y$ is in the scope of a quantifier sequence of the form $Q_1 x \dots Q_2 y$ where x is bound before y . Let $\text{FO}_f(\mathbf{P})$ denote the set of forward formulae over predicates in \mathbf{P} .

Proposition 3.9. $\text{MC}^{\text{URG}}(\text{FO}_f(\rightarrow))$ is undecidable.

Proof. We again reduce the equality problem for reachability sets of two Petri nets N_1 and N_2 . Let \overline{N} be the net presented in Figure 3.1. We propose a forward formula φ so that $\text{Reach}(N_2) = \text{Reach}(N_1)$ if and only if $\text{PURG}(\overline{N}) \models \varphi$:

$$\begin{aligned} \varphi &\stackrel{\text{def}}{=} \forall z_2 \exists z_1 \forall z \exists y_\ell \exists z' (z_2 \rightarrow z) \Rightarrow ((z \rightarrow z') \vee \psi(z_1, z_2, z, y_\ell)) \\ \psi(z_1, z_2, z, y_\ell) &\stackrel{\text{def}}{=} (z_1 \rightarrow z) \wedge (y_\ell \rightarrow z) \wedge ((z_1 \rightarrow y_\ell) \Leftrightarrow \neg(z_2 \rightarrow y_\ell)) \end{aligned}$$

Forward formulae make it harder to quantify over deadlock markings M . Before presenting how formula φ enables the reduction, a short comment on quantification: this formula intends to quantify over z , but the forward constraint imposes first to quantify over z_2 , then on z_1 , and only afterwards on z . This is not a problem since, once z_2 is fixed, variable z_1 may be fixed, and then z may be chosen. The idea of φ is to capture the situation in Figure 3.1, potentially with the roles of M_1 and M_2 swapped. In detail, the formula considers an arbitrary marking M_2 , a corresponding marking M_1 (if it exists), and an arbitrary marking M . If M_2 and M are not connected, then φ requires nothing. If M_2 and M are connected and M is no deadlock, there are also no requirements. Otherwise M_2 and M are connected and M is a deadlock. In this case, there must be a marking M_\circ (valuation for y_ℓ) so that formula ψ is true for (M_1, M_2, M, M_\circ) . The formula ψ checks that deadlock M is reachable in both N_1 and N_2 , see Figure 3.1. Thus, $\text{Reach}(N_1) = \text{Reach}(N_2)$ iff $\text{PURG}(\overline{N}) \models \varphi$. This proves the claimed undecidability. □

Open problem 2. Decidability status of $\text{MC}^{\text{URG}}(\text{FO}_f(\overset{*}{\rightarrow}))$. ○

While forward formulae can well identify the deadlock markings used in the proof schema, the difficulty is in the description of the local environment witnessing the simulation results.

3.3. Taming undecidability with fragments. In this section, we present the restrictions of $\text{FO}(\rightarrow)$ that we found to have decidable model checking or validity problems.

3.3.1. Existential fragment. Our undecidability results follow a common principle, namely identifying a local pattern in the reachability graph that characterizes an undecidable property. The pattern may depend on the specification language. Below, we state a result that, at first glance, might seem to contradict the previous findings: decidability of $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ restricted to the existential fragment. This decidability, however, simply implies that universal quantification is needed to characterize undecidable properties by local patterns. We write $\exists\text{FO}$ for the fragment of FO consisting of those formulae that use only existential quantification when written in prenex normal form.

Proposition 3.10. $\text{MC}^{\text{URG}}(\exists\text{FO}(\rightarrow, =))$ is decidable.

Proof. Let $N = (P, T, F, M_0)$ be a Petri net with reachability set $\text{Reach}(N)$ and $|P| = n$. Decidability follows from two crucial properties:

- (1) Given a Presburger formula $\varphi(\vec{x}_1, \dots, \vec{x}_\alpha)$ with $n \times \alpha$ free variables such that each \vec{x}_i is a sequence of n distinct variables interpreted as a marking of N , one can decide whether $\varphi(M_1, \dots, M_\alpha)$ holds true for some (not necessarily distinct) markings M_1, \dots, M_α in $\text{Reach}(N)$. Proposition 2.3 corresponds to the case $\alpha = 1$.
- (2) One can effectively construct a quantifier-free Presburger formula $\varphi_{\rightarrow}(\vec{x}_1, \vec{x}_2)$ so that for all markings M_1, M_2 , formula $\varphi_{\rightarrow}(M_1, M_2)$ holds iff $M_1[t]M_2$ for some $t \in T$.

Before we turn to the proofs of (1) and (2), we explain how these results yield decidability of $\text{MC}^{\text{URG}}(\exists\text{FO}(\rightarrow, =))$. Consider $\psi = \exists x_1, \dots, x_\alpha \psi'$ where ψ' is a quantifier-free formula with atomic propositions of the form $x_i \rightarrow x_j$ and $x_i = x_j$. With (2), one constructs a quantifier-free Presburger formula $\varphi(\vec{x}_1, \dots, \vec{x}_\alpha)$ so that for all markings M_1, \dots, M_α in $\text{Reach}(N)$, formula $\varphi(M_1, \dots, M_\alpha)$ holds true iff $\text{PURG}(N), \mathbf{v} \models \psi'$ where $\mathbf{v}(\vec{x}_i) = M_i$ for $1 \leq i \leq \alpha$. By (1), it is decidable whether $\varphi(M_1, \dots, M_\alpha)$ holds for some markings $M_1, \dots, M_\alpha \in \text{Reach}(N)$. This is equivalent to $\text{URG}(N) \models \psi$.

It remains to prove (1) and (2). The formula $\varphi_{\rightarrow}(\vec{x}_1, \vec{x}_2)$ for statement (2) encodes the definition of enabledness and firing for transitions, $M[t]M'$:

$$\bigvee_{t \in T} \left(\bigwedge_{p \in P} \vec{x}_1(p) \geq F(p, t) \right) \wedge \left(\bigwedge_{p \in P} \vec{x}_2(p) = \vec{x}_1(p) - F(p, t) + F(t, p) \right).$$

For statement (1), we adapt the proof of Proposition 2.3. We construct a Petri net N' that simulates α copies of N . Technically, N' is defined as the disjoint union of α instances of N . The initial marking of N' is α times M_0 . For all markings M_1, \dots, M_α we now have the following equivalence: the markings are reachable in N and satisfy $\varphi(M_1, \dots, M_\alpha)$ iff (M_1, \dots, M_α) is a possible simulation result in N' and $\varphi(M_1, \dots, M_\alpha)$ holds. An application of Proposition 2.3 on N' and φ yields the desired decidability result. \square

Again, decidability is preserved with Presburger-definable properties on markings and with labelled transition relations of the form \xrightarrow{t} .

Corollary 3.11. $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow, =))$ restricted to Boolean combinations of existential formulae is decidable.

Consequently, the following subgraph isomorphism problem is decidable too:

- input:** a finite directed graph $\mathcal{G} = (V, E)$ and a Petri net N .
question: is there a subgraph of $(\text{Reach}(N), \rightarrow)$ isomorphic to \mathcal{G} ?

Open problem 3. Decidability status of $\text{MC}^{\text{URG}}(\exists\text{FO}(\overset{*}{\rightarrow}))$ and $\text{MC}^{\text{URG}}(\exists\text{FO}(\overset{*}{\rightarrow}, \rightarrow))$. \circ

3.3.2. $\text{ML}(\square)$ with arithmetical constraints. Section 3.2.2 proves that $\text{VAL}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$ is undecidable. To our surprise, and in contrast to the negative result on model checking the forward fragment of FO, this undecidability depends on the backward modality. The following Proposition 3.12 shows decidability of the validity problem for $\text{ML}(\square)$, even in the presence of arithmetical constraints at the atomic level.

Proposition 3.12. The validity problem $\text{VAL}^{\text{URG}}(\text{PAML}(\square))$ is decidable.

Proof. Let N be a Petri net, and φ a formula in $\text{PAML}(\square)$. According to Lemma 3.13 stated hereafter, the set of markings satisfying $\neg\varphi$ is effectively semilinear. Let $X_{\neg\varphi}$ be this set. Proving validity of φ amounts to checking that no element of $X_{\neg\varphi}$ is reachable in N . This is decidable from Proposition 2.3. \square

Lemma 3.13. *Given a Petri net N with n places and a formula φ in $\text{PAML}(\square)$, the set of markings in \mathbb{N}^n satisfying φ in $\text{UG}(N)$ is effectively semilinear.*

Proof. We proceed by induction on the structure of φ , using the fact that semilinear sets are (effectively) closed under Boolean operations and the fact that, if X is semilinear, then $\text{pre}(X) = \{M \in \mathbb{N}^n : \exists M' \in X, M \rightarrow M'\}$ is effectively semilinear too. The latter set $\text{pre}(X)$ contains all markings with a successor marking in X .

Each atomic formula is a quantifier-free Presburger formula, and as such, defines a semilinear set. Throughout the induction on the structure of φ , formulae with outermost Boolean connectives are treated in the obvious way by applying Boolean operations on semilinear sets. Eventually one has to prove that $\square\psi$ defines a semilinear set whenever ψ does. Using the induction hypothesis, let X_ψ be the semilinear set of markings satisfying ψ . The set satisfying $\square\psi$ is then equal to $\mathbb{N}^n \setminus \text{pre}(\mathbb{N}^n \setminus X_\psi)$, which is effectively semilinear. This concludes the induction, and the proof. \square

This decidability result can be extended by allowing labels on edges (transitions).

3.4. On the hardness of decidable problems. Some of our decision procedures call subroutines for checking reachability in Petri nets, even though the reachability problem is not known to be primitive recursive. We provide here some complexity-theoretic justification for these costly invocations: we reduce the reachability problem for Petri nets to the decidable problems $\text{MC}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$ and $\text{MC}^{\text{URG}}(\exists\text{FO}(\rightarrow))$. Besides reachability, we proposed decision procedures that exploit the effective semilinearity of reachability sets or relations (see e.g. Proposition 2.7). The next proposition shows that, already for bounded Petri nets, $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ is of high complexity.

Proposition 3.14. *$\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ restricted to bounded Petri nets is decidable but this problem has nonprimitive recursive complexity.*

Proof. We perform a reduction from the finite containment problem for Petri nets, known to have nonprimitive recursive complexity [36]. Let N_1 and N_2 be two bounded Petri nets with identical sets of places, and construct \overline{N} as in Section 3.1. This net is bounded. The formula φ in $\text{FO}(\rightarrow)$ that checks inclusion is derived from the formula in Section 3.1:

$$\forall z (\neg \exists z' z \rightarrow z') \Rightarrow (\exists z_2 z_2 \rightarrow z \wedge \neg \varphi_l(z_2))$$

where $\varphi_l(x) \stackrel{\text{def}}{=} \exists y (x \rightarrow y \wedge y \rightarrow x)$. The construction guarantees $\text{Reach}(N_1) \subseteq \text{Reach}(N_2)$ iff $\text{URG}(\overline{N}) \models \varphi$. Indeed, a deadlock is either reachable from N_2 or from N_1 . But to satisfy the formula, if the deadlock is reachable from N_1 it also has to be reachable from N_2 . Note that the formula φ is again independent of N_1 and N_2 . \square

We have seen that $\text{VAL}^{\text{URG}}(\text{ML}(\square))$ is decidable by reduction to the reachability problem for Petri nets (see Proposition 3.12). Below, we state that there is a reduction in the reverse direction, from non-reachability to $\text{VAL}^{\text{URG}}(\text{ML}(\square))$.

Proposition 3.15. *There is a logarithmic-space reduction from the non-reachability problem for Petri nets to $\text{VAL}^{\text{URG}}(\text{ML}(\square))$.*

Proof. Without any loss of generality, we can assume that the non-reachability problem is restricted to the target marking $\vec{0}$ (no place has any token). Consider the Petri net $N = (P, T, F, M_0)$ where we assume w.l.o.g. that every transition has a place in its preset.

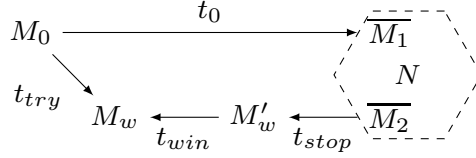


Figure 3.2: Reachability graph in the hardness proof of $\text{ML}(\square, \square^{-1})$ -model checking

We build a variant Petri net N' from N by adding a new transition t_p for every place $p \in P$. The new transitions are put in self-loop with their places, $F'(p, t_p) = 1 = F'(t_p, p)$ and $F'(p', t_p) = 0 = F'(t_p, p')$ for all $p' \in P$ with $p' \neq p$. Intuitively, t_p witness for the presence of tokens on p by the existence of at least one transition from M in the reachability graph. As a result, $\vec{0} \notin \text{Reach}(N)$ iff for every marking $M \in \text{Reach}(N')$, some transition can be fired: $(D, \rightarrow), M \models \diamond \top$. Note that our reduction uses a constant formula. \square

Proposition 3.16. *There is a logarithmic-space reduction from the reachability problem for Petri nets to $\text{MC}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$.*

Proof. We reduce reachability of marking M_2 from marking M_1 in a Petri net N to an instance of $\text{MC}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$ for a larger net \bar{N} . The idea is to introduce a marking M_w (see Figure 3.2) such that the existence of a path to M_w of length greater than 1 is a witness for the existence of some path from M_1 to M_2 in $\text{PURG}(N)$. To reach M_w by an ML formula, we place it close to the new initial marking. We sketch the argumentation. The initial marking M_0 of \bar{N} contains a single marked place p_i for which two transitions t_{try} and t_0 compete. Transition t_{try} moves the unique token from p_i to another place p_w and thus produces the marking M_w where no other place is marked. Transition t_0 loads M_1 in the places of N and moves the control token from p_i to another control place p_c set in self-loop with all transitions of N . This starts the simulation of N from M_1 . The simulation may get stuck or proceed forever, or it may be interrupted whenever it reaches a marking of N greater than or equal to M_2 . Then, transition t_{stop} consumes M_2 from the places of N and moves the control token from p_c to a place $p_{w'}$. The control token is finally moved from $p_{w'}$ to p_w by firing t_{win} . M_w is reached, after firing $t_{stop} t_{win}$, iff \bar{M}_2 is reached. Therefore M_2 is reachable from M_1 iff M_w is reachable from \bar{M}_1 (its restriction to the places of N is M_1). This is equivalent to stating that M_w has a predecessor different from M_0 . The shape of the reachability graph allows us to formulate the latter as a local property in $\text{ML}(\square, \square^{-1})$:

$$\varphi := \diamond(\square \perp \wedge \diamond^{-1} \diamond^{-1} \top).$$

Without loss of generality, we can assume that M_1 is no deadlock and $M_2 \neq M_1$. Formula φ requires that M_0 has a deadlock successor which has an incoming path of length two. That the successor is a deadlock means it is not \bar{M}_1 but M_w obtained by firing t_{try} . The path from M_0 to M_w is of length one and M_0 has no predecessor. So the path of length two to M_w is not via t_{try} but stems from t_{win} . This means M_w is reachable from \bar{M}_1 , which means M_2 is reachable from M_1 in N . \square

The proof of Proposition 3.16 can be adapted to $\exists\text{FO}(\rightarrow)$ for which we also have shown decidability of model-checking by reduction to the reachability problem for Petri nets.

Proposition 3.17. *There is a logarithmic-space reduction from the reachability problem for Petri nets to $\text{MC}^{\text{URG}}(\exists\text{FO}(\rightarrow))$ restricted to a single variable.*

Proof. Among any two of the following problems, there is a logarithmic-space reduction:

- (1) the reachability problem for Petri nets;
- (2) the reachability problem for Petri nets restricted to instances such that the target marking M is equal to $\vec{0}$;
- (3) the following variant of the reachability problem:

input: a Petri net $N = (P, T, F, M_0)$ with no neutral transitions and a place $p \in P$.

question: Is there a marking M with $M(p) = 0$ such that $M \in \text{Reach}(N)$?

To show that (3) is as hard as reachability, the idea is to introduce a budget place that maintains the sum of tokens in all other places. From an instance of problem (3), let us build an instance of $\text{MC}^{\text{URG}}(\exists\text{FO}(\rightarrow))$ restricted to a single variable. We build a Petri net N' from N and $p \in P$ by simply adding a neutral transition (the unique one in N') that is in self-loop with p . One can then easily show that there is a marking $M \in \text{Reach}(N)$ with $M(p) = 0$ iff $\text{PURG}(N') \models \exists x \neg(x \rightarrow x)$. \square

4. FO WITH REACHABILITY PREDICATES

In this section, we consider several first-order languages with reachability relations $\xrightarrow{*}$ or $\xrightarrow{\pm}$, mainly without the one-step relation \rightarrow . Undecidability of these dialects does not directly follow from Theorem 3.3 since we may exclude \rightarrow . Nonetheless we follow the same proof schema. Besides, we distinguish the case when reachability sets are semilinear leading to a surprising undecidability result (Proposition 4.5). Finally, we show that model-checking unlabelled graphs with $\text{FO}(\rightarrow, \xrightarrow{*})$ is undecidable too.

4.1. FO with reachability relations. Let us see why the model checking problem for both the strict and the non-strict reachability relation is undecidable.

4.1.1. Undecidability of $\text{MC}^{\text{URG}}(\text{FO}(\xrightarrow{\pm}))$. The decidability status of $\text{MC}^{\text{URG}}(\text{FO}(\xrightarrow{\pm}))$ is not directly dependent upon the decidability status of $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$. Still we are able to adapt the construction of Section 3.1 but using now a formula φ in $\text{FO}(\xrightarrow{\pm})$. The Petri net \bar{N} is the one depicted on Figure 3.1. The formula φ is defined as follows:

$$\varphi \stackrel{\text{def}}{=} \forall z \, dl(z) \Rightarrow (\exists z_1 (z_1 \xrightarrow{\pm} z) \wedge \varphi_{\text{left}}(z_1)) \wedge (\exists z_2 (z_2 \xrightarrow{\pm} z) \wedge \varphi_{\text{right}}(z_2))$$

where

- * $dl(z) \stackrel{\text{def}}{=} \neg \exists z' z \xrightarrow{\pm} z'$,
- * $sl(y) \stackrel{\text{def}}{=} y \xrightarrow{\pm} y \wedge \forall w [y \xrightarrow{\pm} w \Rightarrow w \xrightarrow{\pm} y]$,
- * $\varphi_{\text{left}}(z) \stackrel{\text{def}}{=} [\exists y z \xrightarrow{\pm} y \wedge sl(y)] \wedge [\forall y z \xrightarrow{\pm} y \Rightarrow (sl(y) \vee dl(y))]$,
- * $\varphi_{\text{right}}(z) \stackrel{\text{def}}{=} [\exists y z \xrightarrow{\pm} y \wedge \forall y z \xrightarrow{\pm} y \Rightarrow dl(y)]$.

Lemma 4.1. $\text{Reach}(N_1) = \text{Reach}(N_2)$ iff $\text{PURG}(\bar{N}) \models \varphi$.

Proof. The principles presented in the proof of Lemma 3.1 apply here. Below, we refer to markings as they are depicted in Figure 3.1.

First, observe that none of the formulae $dl(z)$, $sl(y)$, $\varphi_{\text{left}}(z)$ nor $\varphi_{\text{right}}(z)$ may be satisfied at a marking reached in course of simulating the original Petri nets N_1 or N_2 : the formula $dl(z)$, which asserts the absence of a successor, is always false on such markings

whereas the formula $sl(y)$, requiring that one can always come back to y , is false at such markings since the transitions t_{end}^1 and t_{end}^2 cannot be undone. Furthermore, neither dl nor sl is satisfied by the markings M_1 or M_2 . Hence, formulae $\varphi_{left}(z)$ and $\varphi_{right}(z)$ are not satisfied by any marking z reached in the course of simulating N_1 or N_2 : any such marking has at least one successor of the type M_1 or M_2 , thus invalidating the subformulae $\forall y z \xrightarrow{+} y \Rightarrow (sl(y) \vee dl(y))$ and $\forall y z \xrightarrow{+} y \Rightarrow dl(y)$.

Now, it is straightforward to verify the following facts:

- ★ $dl(z)$ is satisfied precisely at markings M_r and M_ℓ ;
- ★ $sl(y)$ is satisfied precisely at marking M_\circ ;
- ★ φ_{left} and φ_{right} are satisfied respectively at markings M_1 and M_2 .

The formula φ may be written $\forall z \varphi'(z)$ with $\varphi'(z)$ of the form $dl(z) \Rightarrow \psi(z)$. Formula $\varphi'(z)$ is true whenever z evaluates to a non-deadlock marking. Otherwise, when z is a deadlock, validity of ψ requires that it has two distinct predecessors z_1 and z_2 of the types M_1 and M_2 , entailing the equality of the reachability sets of N_1 and N_2 . Conversely, if both reachability sets are equal, then all markings of N_1 and N_2 are connected as described in Figure 3.1, entailing the validity of φ in \overline{N} . \square

Corollary 4.2. $MC^{\text{URG}}(\text{FO}(\xrightarrow{+}))$ is undecidable. Furthermore this results holds for the fixed formula φ defined earlier.

4.1.2. *Undecidability of $MC^{\text{URG}}(\text{FO}(\xrightarrow{*}))$.* For showing undecidability of $MC^{\text{URG}}(\text{FO}(\xrightarrow{*}))$, we have to adapt our usual proof schema since, in $\text{FO}(\xrightarrow{*})$, we are no longer able to identify 1-loops as we did in $\text{FO}(\xrightarrow{+})$. The new schema is illustrated in Figure 4.1.

Proposition 4.3. $MC^{\text{URG}}(\text{FO}(\xrightarrow{*}))$ is undecidable.

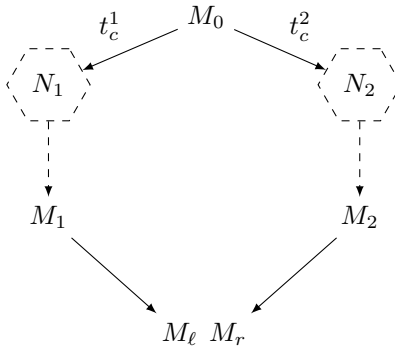


Figure 4.1: Petri net \overline{N} adapted for $\text{FO}(\xrightarrow{*})$

Proof. From two Petri nets N_1 and N_2 , we construct the Petri net \overline{N} depicted in Figure 4.1. We define the following formulae:

- ★ $dl(z) \stackrel{\text{def}}{=} \forall w z \xrightarrow{*} w \Rightarrow w \xrightarrow{*} z$,
- ★ $predl(z) \stackrel{\text{def}}{=} \neg dl(z) \wedge (\forall w (z \xrightarrow{*} w \wedge \neg w \xrightarrow{*} z) \Rightarrow dl(w))$.

Thus in Figure 4.1, the markings M_r and M_ℓ satisfy dl , and the markings M_1 and M_2 satisfy $predl$, but no other marking satisfies these predicates.

The formula φ is defined as follows:

$$\varphi \stackrel{\text{def}}{=} \forall z (dl(z) \Rightarrow \exists z_1, z_2 (z_1 \xrightarrow{*} z \wedge predl(z_1) \wedge z_2 \xrightarrow{*} z \wedge predl(z_2) \wedge \neg z_1 \xrightarrow{*} z_2))$$

Observe that $\neg z_1 \xrightarrow{*} z_2$ ensures that z_1 and z_2 have distinct interpretations. By construction, $\text{Reach}(N_1) = \text{Reach}(N_2)$ iff $\text{PURG}(\overline{N}) \models \varphi$. \square

Even though $\text{MC}^{\text{UG}}(\text{FO}(\rightarrow, =))$ is decidable (see Proposition 2.5), replacing \rightarrow by $\xrightarrow{*}$ and adding $init$ leads to undecidability.

Corollary 4.4. $\text{MC}^{\text{UG}}(\text{FO}(init, \xrightarrow{*}))$ is undecidable.

Indeed, $\text{MC}^{\text{URG}}(\text{FO}(\xrightarrow{*}))$ reduces to $\text{MC}^{\text{UG}}(\text{FO}(init, \xrightarrow{*}))$ by relativization: $\text{URG}(N) \models \varphi$ iff $\text{UG}(N) \models \exists x_0 \text{ init}(x_0) \wedge f(\varphi)$ where φ and $f(\varphi)$ are in $\text{FO}(\xrightarrow{*})$, f is homomorphic for Boolean connectives and $f(\forall x \psi) \stackrel{\text{def}}{=} \forall x (x_0 \xrightarrow{*} x) \Rightarrow f(\psi)$.

Open problem 4. Decidability status of $\text{MC}^{\text{UG}}(\text{FO}(\xrightarrow{*}))$. \circ

4.2. When semilinearity enters into the play. We saw that $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow, =))$ restricted to Petri nets with effectively semilinear reachability sets is decidable, using a translation into Presburger arithmetic (see Proposition 2.7). This section is devoted to discovering what happens when the relation $\xrightarrow{*}$ is added. We establish that $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow, \xrightarrow{*}))$ restricted to Petri nets with semilinear reachability sets is undecidable, by a reduction from $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$. Given a Petri net N and a sentence $\varphi \in \text{FO}(\rightarrow)$, we reduce the truth of φ in $\text{PURG}(N)$ to the truth of a formula $\overline{\varphi}$ in $\text{PURG}(\overline{N})$ where \overline{N} is an augmented Petri net with a semilinear reachability set. The Petri net \overline{N} is defined from N by adding the new places p_0, p_1 and p_2 ; each transition from N is in self-loop with p_1 . Moreover, we add a new set of transitions in self-loop with p_2 , each of which adds tokens to or removes tokens from a corresponding (original) place of N (thus modifying its contents arbitrarily). These transitions form a subnet denoted by Br . Three other transitions are added; see Figure 4.2 for a schematic representation of \overline{N} (the initial marking M'_0 of \overline{N} restricted to places in N is M_0 , while $M'_0(p_0) = M'_0(p_1) = 1$ and $M'_0(p_2) = 0$). Our intention is to force $\text{Reach}(\overline{N})$ to be semilinear while staying able to identify a subset from $\text{Reach}(\overline{N})$ in bijection with $\text{Reach}(N)$; this is a way to drown $\text{Reach}(N)$ into $\text{Reach}(\overline{N})$. Indeed, $\text{Reach}(\overline{N})$ contains all markings such that the sum of p_1 and p_2 is 1 and p_0 is at most 1. Nevertheless, if the transition t is fired first, then the subsequently reachable markings are exactly those of N (except that p_1 contains one token); $\text{PURG}(N)$ embeds isomorphically into $\text{PURG}(\overline{N})$. Until t is fired, one may always come back to M'_0 , using the brownian subnet Br , but this is impossible afterwards.

Proposition 4.5. $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow, \xrightarrow{*}))$ restricted to Petri nets with semilinear reachability sets is undecidable.

Proof. In a first stage, we use $init$ although this predicate cannot be expressed in $\text{FO}(\rightarrow, \xrightarrow{*})$. Let $\overline{\varphi}$ be the formula $\exists x_0 x_1 \text{ init}(x_0) \wedge x_0 \rightarrow x_1 \wedge \neg(x_1 \xrightarrow{*} x_0) \wedge f(\varphi)$ where $f(\cdot)$ is homomorphic for Boolean connectives and $f(\forall x \psi) \stackrel{\text{def}}{=} \forall x (x_1 \xrightarrow{*} x) \Rightarrow f(\psi)$ (relativization). In $\overline{\varphi}$, x_0 is

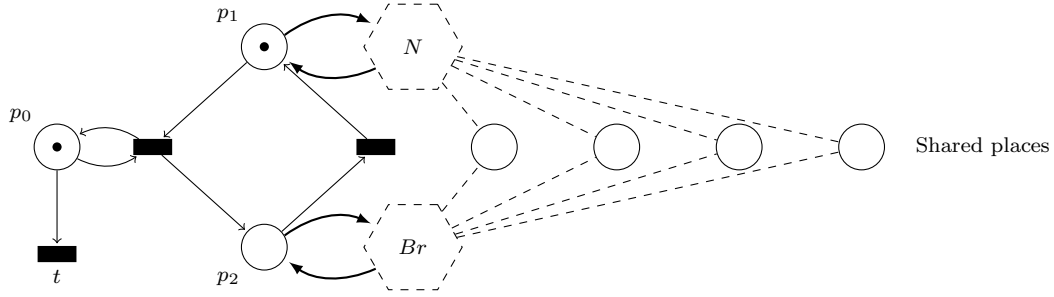


Figure 4.2: Petri net \overline{N}

interpreted as the initial marking M'_0 , and x_1 is interpreted as a successor of x_0 from which x_0 cannot be reached again. This may only happen by firing t from M'_0 . Now the relativization of every other variable to x_1 in $\overline{\varphi}$ ensures that $\text{PURG}(N) \models \varphi$ iff $\text{PURG}(\overline{N}) \models \overline{\varphi}$. To remove *init*, we construct a Petri net \overline{N}' very similar to \overline{N} . \overline{N}' has an extra place p'_0 , initially marked with one token, and a new transition that consumes this token and produces two tokens in p_0 and p_1 , which were initially empty. By construction, the initial marking of \overline{N}' is the sole marking in $\text{PURG}(\overline{N}')$ with no incoming edge and one outgoing edge. With this modified net, we use the modified formula $\overline{\varphi}'$ as follows:

$$\exists x'_0 x_0 x_1 (\neg \exists y y \rightarrow x'_0) \wedge x'_0 \rightarrow x_0 \wedge x_0 \rightarrow x_1 \wedge (\neg x_1 \xrightarrow{*} x_0) \wedge f(\varphi)$$

For the same reasons as above, $\text{PURG}(N) \models \varphi$ iff $\text{PURG}(\overline{N}') \models \overline{\varphi}'$. □

Open problem 5. Decidability status of $\text{MC}^{\text{URG}}(\text{FO}(\xrightarrow{*}))$ restricted to Petri nets with semi-linear reachability sets. ○

4.3. The reachability relation and structure $\text{UG}(N)$. Corollary 4.4 has stated a first undecidability result for the structure $\text{UG}(N)$. In this section, we examine two other situations where it is an undecidable problem to model-check formulas of $\text{FO}(\rightarrow, \xrightarrow{*})$ in $\text{UG}(N)$.

Proposition 4.6. $\text{MC}^{\text{UG}}(\text{FO}(\rightarrow, \xrightarrow{*}))$ is undecidable.

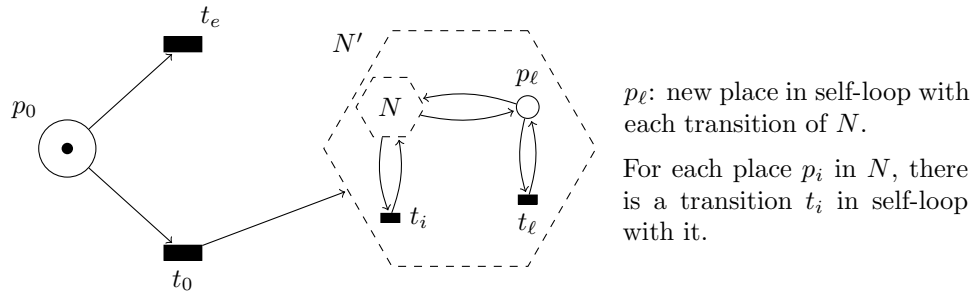


Figure 4.3: Petri net \overline{N}

Proof. We reduce $\text{MC}^{\text{URG}}(\text{FO}(\overset{*}{\rightarrow}))$ to $\text{MC}^{\text{UG}}(\text{FO}(\rightarrow, \overset{*}{\rightarrow}))$. Given a net $N = (P, T, F, M_0)$ and a formula φ in $\text{FO}(\overset{*}{\rightarrow})$, we construct \overline{N} and φ' such that $\text{URG}(N) \models \varphi$ iff $\text{UG}(\overline{N}) \models \varphi'$. Figure 4.3 presents some key elements for the construction of \overline{N} .

First, let $N' = (P', T', F', M'_0)$ be the Petri net defined with $P' = P \cup \{p_\ell\}$, $T' = T \cup \{t_i \mid p_i \in P'\}$, for all (p, t) in $P \times T$, $F'(p, t) = F(p, t)$ and $F'(t, p) = F(t, p)$, for all $p_i \in P'$, $F(p_i, t_i) = F(t_i, p_i) = 1$, for all $t \in T$, $F(t, p_\ell) = F(p_\ell, t) = 1$, for all $p \in P$, $M'_0(p) = M_0(p)$, and $M'_0(p_\ell) = 1$. Restricted to places in P (all places but p_ℓ), the reachable markings of N' coincide with those of N . By construction, p_ℓ contains always a single token. In $\text{URG}(N')$, every marking has a 1-loop. Similarly, every marking of N' in which some place is positive possesses a 1-loop in the graph $\text{UG}(N')$. The tuple $(0, 0, \dots, 0)$, on the other hand, enables no transition (the empty place p_ℓ inhibits every transition).

Now, we construct \overline{N} from N' . \overline{N} has the same places and transitions as N' , plus an extra place p_0 and two extra transitions t_e and t_0 . Transition t_e removes tokens from p_0 , one at a time. Transition t_0 consumes one token from p_0 and produces M'_0 in the places of N' . The initial marking \overline{M}_0 of \overline{N} has a single token in place p_0 .

We claim the following:

- ★ The reachable graph of \overline{N} is identical to the reachable graph of N , up to the first transition and up to the 1-loops which have no influence on formulas in $\text{FO}(\overset{*}{\rightarrow})$.
- ★ There is a formula $\varphi_{\text{init}}(x) \in \text{FO}(\rightarrow, \overset{*}{\rightarrow})$ which is satisfied in $\text{UG}(\overline{N})$ only at \overline{M}_0 .

Assuming these claims, validity of a formula in $\text{FO}(\overset{*}{\rightarrow})$ with respect to $\text{URG}(N)$ may be reduced to the validity of a formula of $\text{FO}(\rightarrow, \overset{*}{\rightarrow})$ with respect to $\text{UG}(\overline{N})$, using a similar technique as in the proof of Corollary 4.4. For this purpose, we should relativize the given formula in $\text{FO}(\overset{*}{\rightarrow})$ to the vertices of $\text{UG}(\overline{N})$ that may be reached from the marking \overline{M}_0 defined by $\overline{M}_0[t_0]\overline{M}'_0$. This can actually be done in $\text{FO}(\rightarrow, \overset{*}{\rightarrow})$, because \overline{M}'_0 is the sole marking of \overline{N} that satisfies the formula $\exists y \varphi_{\text{init}}(y) \wedge y \rightarrow x \wedge x \rightarrow y$. Therefore, to complete the proof of the proposition, it suffices to establish the two claims made above.

Now, the first claim derives immediately from the construction of \overline{N} . The second claim may be established by setting:

$$\varphi_{\text{init}}(x) \stackrel{\text{def}}{=} (\neg x \rightarrow x) \wedge (\exists y \forall z x \rightarrow y \wedge \neg(y \rightarrow z))$$

This formula contains a subformula $(\neg x \rightarrow x)$ that expresses the absence of a 1-loop, thus $\varphi_{\text{init}}(x)$ may only be satisfied in markings with all places $p \in P'$ empty. But $(\neg x \rightarrow x)$ may be satisfied in a marking x with an arbitrary number of tokens in p_0 . Now consider markings with all places in P' empty, and an arbitrary number of tokens in p_0 . Three cases must be considered. First, suppose that p_0 contains a single token (i.e., x is interpreted by \overline{M}_0), then $(\exists y \forall z x \rightarrow y \wedge \neg(y \rightarrow z))$ is satisfied: x has a successor y (reached by firing t_e) which is a deadlock. Second, if p_0 is empty, then the marking x has no successor at all. If p_0 contains at least two tokens, then no successor of x is a deadlock: every marking reached by t_0 has a 1-loop and t_e can be executed at least twice. Putting everything together, the only tuple in \mathbb{N}^n satisfying $\varphi_{\text{init}}(x)$, is the marking $\overline{M}_0 = (1, 0, \dots, 0)$, establishing the second claim. □

Proposition 4.6 holds even when the reachability set of the net is effectively semilinear.

Proposition 4.7. $\text{MC}^{\text{UG}}(\text{FO}(\rightarrow, \overset{*}{\rightarrow}))$ is undecidable for the subclass of Petri nets with an effective semilinear reachability set.

Proof. We pile up (adaptations of) the proofs of Propositions 4.3, 4.5, and 4.6.

Given arbitrary two nets N_1 and N_2 without neutral transitions, let N_3 denote the net \overline{N} constructed from N_1 and N_2 like in the proof of Proposition 4.3, and let M_3 denote the initial marking of this net. By the proof of Proposition 4.3, $\text{Reach}(N_1) = \text{Reach}(N_2)$ if and only if $\text{PURG}(N_3) \models \varphi$, where:

$$\begin{aligned} \varphi &\stackrel{\text{def}}{=} \forall z (dl(z) \Rightarrow \exists z_1, z_2 (z_1 \xrightarrow{*} z \wedge \text{predl}(z_1) \wedge z_2 \xrightarrow{*} z \wedge \text{predl}(z_2) \wedge \neg z_1 \xrightarrow{*} z_2 \wedge \neg z_2 \xrightarrow{*} z_1)), \\ dl(z) &\stackrel{\text{def}}{=} \forall w z \xrightarrow{*} w \Rightarrow w \xrightarrow{*} z, \\ \text{predl}(z) &\stackrel{\text{def}}{=} \neg dl(z) \wedge (\forall w)(z \xrightarrow{*} w \wedge \neg w \xrightarrow{*} z) \Rightarrow dl(w). \end{aligned}$$

Let $\text{PURG}_{\circlearrowleft}(N_3)$ be the extended reachability graph obtained from $\text{PURG}(N_3)$ by adding a 1-loop in every marking. Then clearly, $\text{PURG}(N_3) \models \varphi$ if and only if $\text{PURG}_{\circlearrowleft}(N_3) \models \varphi$. By Hack's result, $\text{PURG}_{\circlearrowleft}(N_3) \models \varphi$ is undecidable from the input $\{N_1, N_2\}$.

Now put $N = N_3$ in the net shown in Figure 4.2. Denote the resulting net \overline{N} by N_4 , and let M_4 be its initial marking. By construction, N_4 has a semilinear reachability set. Moreover, if we put:

$$\theta(x, y) \stackrel{\text{def}}{=} \text{init}(x) \wedge x \rightarrow y \wedge \neg(y \xrightarrow{*} x),$$

then, in $\text{PURG}(N_4)$, this statement holds exclusively for x interpreted by M_4 and y interpreted by $M_3 + \{p_1\}$. Let $\text{PURG}_{\circlearrowleft}(N_4)$ be the extended reachability graph obtained from $\text{PURG}(N_4)$ by adding a 1-loop in every marking. Then clearly, in $\text{PURG}_{\circlearrowleft}(N_4)$, $\theta(x, y)$ holds exclusively for x interpreted by M_4 and y interpreted by $M_3 + \{p_1\}$.

Finally put $N = N_4$ in the net shown in Figure 4.3. Denote the resulting net \overline{N} by N_5 , and let M_5 be its initial marking. Thus, N_5 has a semilinear reachability set. As was shown in the proof of Proposition 4.6, if we put:

$$\varphi_{\text{init}}(x) \stackrel{\text{def}}{=} (\neg x \rightarrow x) \wedge (\exists y \forall z x \rightarrow y \wedge \neg(y \rightarrow z)),$$

then, in $\text{UG}(N_5)$, $\varphi_{\text{init}}(x)$ holds exclusively for x interpreted by M_5 . Therefore, if we put:

$$\psi_{\text{init}}(x) \stackrel{\text{def}}{=} \exists y \varphi_{\text{init}}(y) \wedge y \rightarrow x \wedge x \rightarrow x,$$

then, in $\text{UG}(N_5)$, $\psi_{\text{init}}(x)$ holds exclusively for x interpreted by $M_4 + \{p_\ell\}$. The subgraph of $\text{UG}(N_5)$ reachable from the marking $M_4 + \{p_\ell\}$ is isomorphic to $\text{PURG}_{\circlearrowleft}(N_4)$. Therefore, in $\text{UG}(N_5)$, $\psi_{\text{init}}(x) \wedge x \rightarrow y \wedge \neg(y \xrightarrow{*} x)$ holds for x, y if and only if x is interpreted by $M_4 + \{p_\ell\}$ and y is interpreted by $M_3 + \{p_1\} + \{p_\ell\}$. The subgraph of $\text{UG}(N_5)$ reachable from the marking $M_3 + \{p_1\} + \{p_\ell\}$ is isomorphic to $\text{PURG}_{\circlearrowleft}(N_3)$. Therefore, $\text{PURG}_{\circlearrowleft}(N_3) \models \varphi$ if and only if $\text{UG}(N_5) \models \overline{\varphi}$ where $\overline{\varphi}$ is the formula:

$$\exists x_0 x_1 \psi_{\text{init}}(x_0) \wedge x_0 \rightarrow x_1 \wedge \neg(x_1 \xrightarrow{*} x_0) \wedge f(\varphi)$$

where $f(\cdot)$ is homomorphic for Boolean connectives and $f(\forall x \psi) \stackrel{\text{def}}{=} \forall x (x_1 \xrightarrow{*} x) \Rightarrow f(\psi)$ (relativization). As a consequence, $\text{UG}(N_5) \models \overline{\varphi}$ is undecidable from the input $\{N_1, N_2\}$. \square

Problem	#	Arbitrary	Effectively semilinear Reach(N)
$\text{MC}^\sharp(\text{FO}(\rightarrow))$	<i>URG</i>	UNDEC (Cor. 3.2)	DEC
$\text{MC}^\sharp(\text{FO}(\overset{\pm}{\rightarrow}))$	<i>URG</i>	UNDEC (Cor. 4.2)	open
$\text{MC}^\sharp(\text{FO}(\overset{*}{\rightarrow}))$	<i>URG</i>	UNDEC (Prop. 4.3)	open
$\text{MC}^\sharp(\text{FO}(\rightarrow, \overset{*}{\rightarrow}))$	<i>URG</i> <i>UG</i>	UNDEC UNDEC (Prop. 4.6)	UNDEC (Prop. 4.5) UNDEC (Prop. 4.7)
$\text{MC}^\sharp(\text{FO}^+(\rightarrow))$	<i>URG</i>	UNDEC (Prop. 3.8)	DEC
$\text{MC}^\sharp(\text{FO}_f(\rightarrow))$	<i>URG</i>	UNDEC (Prop. 3.9)	DEC
$\text{MC}^\sharp(\exists\text{FO}(\rightarrow, =))$	<i>URG</i>	DEC [†] (Prop. 3.10)	DEC
$\text{MC}^\sharp(\text{FO}(\rightarrow))$ with 1 variable	<i>URG</i>	DEC [†] (Prop. 3.17)	DEC
$\text{MC}^\sharp(\text{FO}(\rightarrow, =))$	<i>UG</i>	DEC (Prop. 2.5)	DEC
$\text{MC}^\sharp(\text{ML}(\square))$	<i>URG</i>	PSPACE-complete	PSPACE-complete
$\text{MC}^\sharp(\text{ML}(\square, \square^{-1}))$	<i>URG</i>	DEC [†] (Prop. 2.9)	DEC
$\text{VAL}^\sharp(\text{ML}(\square, \square^{-1}))$	<i>URG</i>	UNDEC (Prop. 3.7)	DEC
$\text{VAL}^\sharp(\text{PAML}(\square))$	<i>URG</i>	DEC [†] (Prop. 3.12)	DEC

Table 1: Summary (†: equivalent to Petri nets (non) reachability problem)

In this section we have examined several first-order sublanguages involving the reachability predicate. We obtained undecidability results, even when the reachable markings form a semilinear set, and even when the global structure $\text{UG}(N)$ is considered instead of $\text{URG}(N)$.

5. CONCLUDING REMARKS

We investigated mainly the model-checking problem over unlabelled reachability graphs of Petri nets with the first-order language $\text{FO}(\rightarrow)$ (no label on transitions, no property on markings). The robustness of our main undecidability proof has been tested against standard fragments of $\text{FO}(\rightarrow)$ (for instance the two-variable fragment), modal fragments from $\text{ML}(\square, \square^{-1})$ and against the additional assumption that reachability sets are effectively semilinear. Table 1 provides a summary of the main results (observe that whenever the reachability relation $\overset{*}{\rightarrow}$ is effectively semilinear, each problem is decidable). Results in bold are proved in the paper, whereas unbold ones are their consequences; furthermore each undecidability result holds for a fixed formula. We have investigated several types of borderlines to distinguish decidable problems from undecidable ones. For instance, $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ restricted to the two-variable fragment is undecidable whereas $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ restricted to the existential fragment is decidable (even though this problem is at least as hard as the reachability problem for Petri nets). Similarly, on the modal side, $\text{MC}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$ is decidable (again as hard as the reachability problem for Petri nets) whereas $\text{VAL}^{\text{URG}}(\text{ML}(\square, \square^{-1}))$ is undecidable. Despite the numerous results we obtained, we can identify the following rules of thumb.

- (1) Undecidability of $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ is robust for numerous fragments of $\text{FO}(\rightarrow)$ including both universal and existential quantifications (a single alternation is enough).
- (2) Decidability results with simple restrictions such as considering bounded Petri nets or $\exists\text{FO}(\rightarrow)$ lead to computationally difficult problems, some of them being non primitive recursive or as hard as the reachability problem for Petri nets (see Section 3.4).
- (3) The above points are still relevant for modal languages.

Let us conclude the paper by mentioning possible continuations of this work. A first direction would be to investigate the model checking of fragments of *second-order languages* with respect to Petri net unlabelled reachability graphs. Knowing that $\text{MC}^{\text{URG}}(\text{FO}(\rightarrow))$ is already undecidable, this makes sense only if one disallows first-order quantification, while keeping of course second-order quantification. A possible primitive atomic formula could be for instance: $X \Longrightarrow Y \stackrel{\text{def}}{\iff}$ for all $x \in X$, there is $y \in Y$ such that $x \rightarrow y$ and for all $y \in Y$, there is $x \in X$ such that $x \rightarrow y$. With this definition, it is easily shown that $\text{MC}^{\text{URG}}(\text{MSO}(\Rightarrow))$ is undecidable, but many other fragments of MSO are worth investigating and comparing with the fragments considered in the paper.

A second direction for extending this work would be to consider the geometrical properties of the set of markings reachable from a given marking, taken as a subset of \mathbb{N}^n . It is for instance trivial to determine whether there is at least one marking reachable from the initial marking and different from it. It is slightly more difficult to prove that there is at least one non-reachable marking.

A third direction, diverging significantly from our approach, would be to investigate decidability questions about infinite *unfoldings* of nets instead of net reachability graphs. Unfolding Petri nets produces *local event structures* that induce in turn *local trace languages* [24]. Safe Petri nets, as opposed to unbounded Petri nets, may in particular be modelled with *regular* trace event structures [34]. The decidability of FO over regular trace event structures has been shown in [34], as well as the decidability of MTL, a fragment of MSO where quantification is restricted to conflict-free sets of events. The proofs of these results rely strongly on regularity and do not extend easily to local event structures representing general Petri nets.

REFERENCES

- [1] T. Araki and T. Kasami. Some decision problems related to the reachability problem for Petri nets. *Theoretical Computer Science*, 3:85–104, 1976.
- [2] T. Araki and T. Kasami. Decidability problems on the strong connectivity of Petri net reachability sets. *Theoretical Computer Science*, 4:99–119, 1977.
- [3] M. F. Atig and P. Habermehl. On Yen’s path logic for Petri nets. *International Journal of Foundations of Computer Science*, 22(4):783–799, 2011.
- [4] H. G. Baker. Rabin’s proof of the undecidability of the reachability set inclusion problem of vector addition systems, 1973. M.I.T., Project MAC, CSGM 1979.
- [5] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. CUP, 2001.
- [6] A. Blumensath and E. Grädel. Automatic structures. In *LICS’00*, pages 51–62, 2000.
- [7] B. Boigelot. *Symbolic methods for exploring infinite state spaces*. PhD thesis, Université de Liège, 1998.
- [8] A. Boudet and H. Comon. Diophantine equations, presburger arithmetic and finite automata. In *CAAP*, volume 1059 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 1996.
- [9] Z. Bouziane and A. Finkel. Cyclic Petri net reachability sets are semi-linear effectively constructible. In *INFINITY’97*, volume 9 of *ENTCS*, pages 15–24. Elsevier, 1997.
- [10] M. Bozga, R. Iosif, and F. Konečný. Fast acceleration of ultimately periodic relations. In *CAV’10*, volume 6174 of *Lecture Notes in Computer Science*, pages 227–242. Springer, 2010.
- [11] H. Comon and Y. Jurski. Multiple counter automata, safety analysis and Presburger analysis. In *CAV’98*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 1998.
- [12] J. Esparza. Petri nets, commutative context-free grammars, and basic parallel processes. *Fundamenta Informaticae*, 31(13):13–26, 1997.
- [13] J. Esparza. Decidability and complexity of Petri net problems — an introduction. In *Advances in Petri Nets 1998*, volume 1491 of *Lecture Notes in Computer Science*, pages 374–428. Springer, 1998.
- [14] A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS’02*, volume 2256 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2002.

- [15] D. Gabbay. Expressive functional completeness in tense logic. In *Aspects of Philosophical Logic*, pages 91–117. Reidel, 1981.
- [16] S. Ginsburg and E. Spanier. Bounded ALGOL-like languages. *Transactions of the American Mathematical Society*, pages 333–368, 1964.
- [17] S. Ginsburg and E. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16:285–296, 1966.
- [18] J. Grabowski. The decidability of persistence for vector addition systems. *Information Processing Letters*, 11:20–23, 1980.
- [19] P. Habermehl. On the complexity of the linear-time mu-calculus for Petri nets. In *ICATPN'97*, volume 1248 of *Lecture Notes in Computer Science*, pages 102–116. Springer, 1997.
- [20] M. Hack. *Decidability Questions for Petri nets*. PhD thesis, MIT, 1975.
- [21] M. Hack. The equality problem for vector addition systems is undecidable. *Theoretical Computer Science*, 2:77–96, 1976.
- [22] D. Hauschildt. Semilinearity of the reachability set is decidable for Petri nets. Technical Report FBI-HH-B-146/90, University of Hamburg, 1990.
- [23] M. Hennessy and R. Milner. On observing nondeterminism and concurrency. In *ICALP'80*, volume 85 of *Lecture Notes in Computer Science*, pages 299–309. Springer, 1980.
- [24] P. W. Hoogers, H. C. M. Kleijn, and P. S. Thiagarajan. An event structure semantics for general Petri nets. *Theoretical Computer Science*, 153:129–170, 1993.
- [25] J. Hopcroft and J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.
- [26] R. Howell, P. Jančar, and L. Rosier. Completeness results for single-path Petri nets. *Information & Computation*, 106(2):253–265, 1993.
- [27] O. Ibarra. Reversal-bounded multicounter machines and their decision problems. *Journal of the ACM*, 25(1):116–133, 1978.
- [28] P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148:281–301, 1995.
- [29] R. M. Karp and R. E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3:147–195, 1969.
- [30] R. Kosaraju. Decidability of reachability in vector addition systems. In *STOC'82*, pages 267–281, 1982.
- [31] J. Leroux. Vector Addition System Reachability Problem (A Short Self-Contained Proof). In *POPL'11*, pages 307–316, 2011.
- [32] J. Leroux. Vector addition system reversible reachability problem. In *CONCUR'11*, volume 6901 of *Lecture Notes in Computer Science*, pages 327–341. Springer, 2011.
- [33] J. Leroux and G. Sutre. On Flatness for 2-Dimensional Vector Addition Systems with States. In *CONCUR'04*, volume 3170 of *Lecture Notes in Computer Science*, pages 402–416. Springer, 2004.
- [34] P. Madhusudan. Model-checking trace event structures. In *LICS'03*, pages 371–380, 2003.
- [35] E. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal of Computing*, 13(3):441–460, 1984.
- [36] E. Mayr and A. Meyer. The complexity of the finite containment problem for Petri nets. *Journal of the ACM*, 28(3):561–576, 1981.
- [37] E. Pelz. Closure properties of deterministic Petri nets. In *STACS'87*, volume 247 of *Lecture Notes in Computer Science*, pages 371–382. Springer, 1987.
- [38] M. Praveen. Small vertex cover makes petri net coverability and boundedness easier. In V. Raman and S. Saurabh, editors, *IPEC*, volume 6478 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2010.
- [39] M. Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du premier congrès de mathématiciens des Pays Slaves, Warszawa*, pages 92–101, 1929.
- [40] S. Schulz. First-order logic with reachability predicates on infinite systems. In *FST&TCS'10*, pages 493–504. LIPICS, 2010.
- [41] R. Valk and G. Vidal-Naquet. Petri nets and regular languages. *Journal of Computer and System Sciences*, 23:299–325, 1981.