Acceleration for Petri Nets

Jérôme Leroux

Acceleration For Petri Nets

Jérôme Leroux*

LaBRI, Université de Bordeaux, CNRS

Abstract. The reachability problem for Petri nets is a central problem of net theory. The problem is known to be decidable by inductive invariants definable in the Presburger arithmetic. When the reachability set is definable in the Presburger arithmetic, the existence of such an inductive invariant is immediate. However, in this case, the computation of a Presburger formula denoting the reachability set is an open problem. Recently this problem got closed by proving that if the reachability set of a Petri net is definable in the Presburger arithmetic, then the Petri net is flat, i.e. its reachability set can be obtained by runs labeled by words in a bounded language. As a direct consequence, classical algorithms based on acceleration techniques effectively compute a formula in the Presburger arithmetic denoting the reachability set.

1 Introduction

Petri Nets are one of the most popular formal methods for the representation and the analysis of parallel processes [1]. The reachability problem is central since many computational problems (even outside the realm of parallel processes) reduce to this problem. Sacerdote and Tenney provided in [14] a partial proof of decidability of this problem. The proof was completed in 1981 by Mayr [13] and simplified by Kosaraju [8] from [13, 14]. Ten years later [9], Lambert provided a further simplified version based on [8]. This last proof still remains difficult and the upper-bound complexity of the corresponding algorithm is just known to be non-primitive recursive. Nowadays, the exact complexity of the reachability problem for Petri nets is still an open-question. Even an Ackermannian upper bound is open (this bound holds for Petri nets with finite reachability sets [2]).

Basically, a Petri net is a pair $(T, \mathbf{c}_{\text{init}})$ where $T \subseteq \mathbb{N}^d \times \mathbb{N}^d$ is a finite set of *transitions*, and $\mathbf{c}_{\text{init}} \in \mathbb{N}^d$ is the *initial configuration.* A vector $\mathbf{c} \in \mathbb{N}^d$ is called a *configuration.* Given a transition $t = (\mathbf{p}, \mathbf{q})$, we introduce the binary relations $\xrightarrow{t}$ over the configurations defined by $\mathbf{x} \xrightarrow{t} \mathbf{y}$ if there exists $\mathbf{v} \in \mathbb{N}^d$ such that $\mathbf{x} = \mathbf{p} + \mathbf{v}$ and $\mathbf{y} = \mathbf{q} + \mathbf{v}$. Notice that in this case $\mathbf{y} - \mathbf{x}$ is the vector $\mathbf{q} - \mathbf{p}$. This vector is called the *displacement* of t, and it is denoted by $\Delta(t)$. Let $\sigma = t_1 \ldots t_k$ be a word of transitions $t_j \in T$. We denote by $\Delta(\sigma) = \sum_{j=1}^{k} \Delta(t_j)$, the *displacement* of σ. We introduce the binary relation $\xrightarrow{\sigma}$ over the configurations defined by $\mathbf{x} \xrightarrow{\sigma} \mathbf{y}$ if there exists a sequence $\mathbf{c}_0, \ldots, \mathbf{c}_k$ of configurations such that $\mathbf{c}_0 = \mathbf{x}$, $\mathbf{c}_k = \mathbf{y}$, and such that $\mathbf{c}_{j-1} \xrightarrow{t_j} \mathbf{c}_j$ for every $1 \leq j \leq k$. A configuration

* Work funded by ANR grant REACHARD-ANR-11-BS02-001.

$\mathbf{c} \in \mathbb{N}^d$ is said to be *reachable* if there exists a word $\sigma \in T^*$ such that $\mathbf{c}_{\text{init}} \xrightarrow{\sigma} \mathbf{c}$. The *reachability set* of a Petri net is the set of reachable configurations.

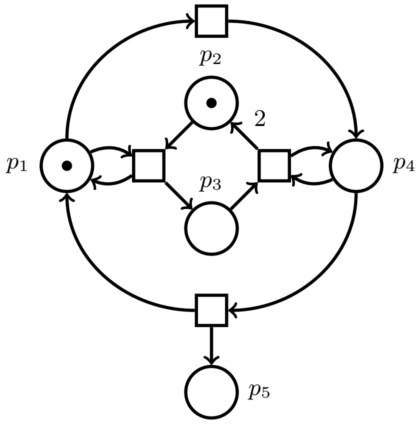

Fig. 1. The Hopcroft and Pansiot net.

Example 1.1. The Petri net depicted in Figure 1 was introduced in [7] as an example of Petri net having a reachability set which cannot be defined by a formula in the logic $\mathrm{FO}\,(\mathbb{N}, +)$, called the *Presburger arithmetic*. In fact, the reachability set is equal to:

$$\left\{ (p_1, p_2, p_3, p_4, p_5) \in \mathbb{N}^5 \mid \begin{array}{l} (\ p_1 = 1 \wedge p_4 = 0 \wedge 1 \leq p_2 + p_3 \leq 2^{p_5}\ ) \vee \\ (\ p_1 = 0 \wedge p_4 = 1 \wedge 1 \leq p_2 + 2p_3 \leq 2^{p_5+1}\ ) \end{array} \right\}$$

Recently, in [10], the reachability sets of Petri nets were proved to be *almost semilinear*, a class of sets that extends the class of Presburger sets (the sets definable in $\mathrm{FO}\,(\mathbb{N}, +)$) inspired by the *semilinear sets* [5]. Note that in general reachability sets are not definable in the Presburger arithmetic [7] (see Example 1.1). An application of the almost semilinear sets was provided; a final configuration is not reachable from an initial one if and only if there exists a forward inductive invariant definable in the Presburger arithmetic that contains the initial configuration but not the final one. Since we can decide if a Presburger formula denotes a forward inductive invariant, we deduce that there exist checkable certificates of non-reachability in the Presburger arithmetic. In particular, there exists a simple algorithm for deciding the general Petri net reachability problem based on two semi-algorithms. A first one that tries to prove the reachability by enumerating finite sequences of actions and a second one that tries to prove the non-reachability by enumerating Presburger formulas. Such an algorithm always terminates in theory but in practice an enumeration does not provide an efficient way for deciding the reachability problem. In particular the problem of deciding *efficiently* the reachability problem is still an *open question.*

When the reachability set is definable in the Presburger arithmetic, the existence of checkable certificates of non-reachability in the Presburger arithmetic is immediate since the reachability set is a forward inductive invariant (in fact the most precise one). The problem of deciding if the reachability set of a Petri is definable in the Presburger arithmetic was studied twenty years ago independently by Dirk Hauschildt during his PhD [6] and Jean-Luc Lambert. Unfortunately, these two works were never published. Moreover, from these works, it is difficult to deduce a simple algorithm for computing a Presburger formula denoting the reachability set when such a formula exists.

For the class of *flat* Petri nets [3, 12], such a computation can be performed with *accelerations techniques.* A Petri net $(T, \mathbf{c}_{\text{init}})$ is said to be *flat* if there exist some words $\sigma_1, \ldots, \sigma_k \in T^*$ such that for every reachable configuration $\mathbf{c}$, there exists a word $\sigma \in \sigma_1^* \ldots \sigma_k^*$ such that $\mathbf{c}_{\text{init}} \xrightarrow{\sigma} \mathbf{c}$. (A language included in $\sigma_1^* \ldots \sigma_k^*$ is said to be *bounded* [4]). *Acceleration techniques* provide a framework for deciding reachability properties that works well in practice but without termination guaranty in theory. Intuitively, acceleration techniques consist in computing with some symbolic representations transitive closures of sequences of actions. For Petri nets, the Presburger arithmetic is known to be expressive enough for this computation. In fact, denoting by $\xrightarrow{\sigma^*}$ the binary relation $\bigcup_{n \in \mathbb{N}} \xrightarrow{\sigma^n}$ where $\sigma \in T^*$, the following lemma shows that $\xrightarrow{\sigma^*}$ can be denoted by a formula in the Presburger arithmetic.

Lemma 1.2 ([3]). *For every word $\sigma \in T^*$ and $n \geq 1$, we have $\mathbf{x} \xrightarrow{\sigma^n} \mathbf{y}$ if, and only if, the following formula holds:*

$$\exists \mathbf{x}', \mathbf{y}' \quad \mathbf{x} \xrightarrow{\sigma} \mathbf{x}' \ \wedge \ \mathbf{y} - \mathbf{x} = n\Delta(\sigma) \ \wedge \ \mathbf{y}' \xrightarrow{\sigma} \mathbf{y}$$

As a direct consequence, since the Presburger arithmetic is a decidable logic, the following algorithm can be implemented by denoting the sets $\mathbf{C}$ with Presburger formulas.

Acceleration$(T, \mathbf{c}_{\text{init}})$
(1) $\mathbf{C} \leftarrow \{\mathbf{c}_{\text{init}}\}$
(2) **while** there exists $\mathbf{c} \xrightarrow{t} \mathbf{c}'$ with $\mathbf{c} \in \mathbf{C}$, $t \in T$ and $\mathbf{c}' \notin \mathbf{C}$
(3) **select** $\sigma \in T^*$
(4) $\mathbf{C} \leftarrow \{\mathbf{y} \in \mathbb{N}^d \mid \exists \mathbf{c} \in \mathbf{C} \ \ \mathbf{c} \xrightarrow{\sigma^*} \mathbf{y}\}$
(5) **return** $\mathbf{C}$

Naturally, when this algorithm terminates, it returns the reachability set. Moreover, under a fairness condition on line (3), this algorithm terminates on any flat Petri net. Basically, it is sufficient to assume that the infinite sequence of words $\sigma_1, \sigma_1, \ldots$, selected during repeated executions of line (3), contains, as subsequences, all the finite sequences of words in T^*. As a direct consequence flat Petri nets have reachability sets effectively definable in the Presburger arithmetic [12]. Recently, we proved that many classes of Petri nets with known

Presburger reachability sets are flat [12], and we conjectured that Petri nets with reachability sets definable in the Presburger arithmetic are flat. In fact, the following theorem shows that the conjecture is true. As a direct consequence, classical tools implementing the previous acceleration algorithms always terminate on the computation of Presburger formulas denoting reachability sets of Petri nets when such a formula exists.

Theorem 1.3 ([11]). *A Petri net is flat if, and only if, its reachability set is definable in the Presburger arithmetic.*

References

1. Javier Esparza and Marcus Nielsen. Decidability issues for petri nets - a survey. *Bulletin of the European Association for Theoretical Computer Science*, 52:245–262, 1994.
2. Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and primitive-recursive bounds with dickson's lemma. In *Proc. of LICS 2011*, pages 269–278. IEEE Computer Society, 2011.
3. Laurent Fribourg. Petri nets, flat languages and linear arithmetic. In María Alpuente, editor, *Proc. of WFLP'2000*, pages 344–365, 2000.
4. S. Ginsburg and E. H. Spanier. Bounded regular sets. *Proceedings of the American Mathematical Society*, 17(5):1043–1049, 1966.
5. Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.
6. Dirk Hauschildt. *Semilinearity of the Reachability Set is Decidable for Petri Nets.* PhD thesis, University of Hamburg, 1990.
7. John E. Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoritical Computer Science*, 8:135–159, 1979.
8. S. Rao Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *Proc. of STOC'82*, pages 267–281. ACM, 1982.
9. Jean Luc Lambert. A structure to decide reachability in petri nets. *Theoretical Computer Science*, 99(1):79–104, 1992.
10. Jérôme Leroux. The general vector addition system reachability problem by Presburger inductive invariants. In *Proc. of LICS 2009*, pages 4–13. IEEE Computer Society, 2009.
11. Jérôme Leroux. Presburger vector addition systems. In *Proc. LICS 2013*. IEEE Computer Society, 2013. To appear.
12. Jérôme Leroux and Grégoire Sutre. Flat counter automata almost everywhere! In *Proc. of ATVA'05*, volume 3707 of *LNCS*, pages 489–503. Springer, 2005.
13. Ernst W. Mayr. An algorithm for the general petri net reachability problem. In *Proc. of STOC'81*, pages 238–246. ACM, 1981.
14. George S. Sacerdote and Richard L. Tenney. The decidability of the reachability problem for vector addition systems (preliminary version). In *Proc. of STOC'77*, pages 61–76. ACM, 1977.