Barylska K, Koutny M, Mikulski L, Piatkowski M. Reversible Computation vs. Reversibility in Petri Nets. ***In: 8th Conference on Reversible Computation*****. 2016, Bologna, Italy: Springer.**

DOI link to article:

http://dx.doi.org/10.1007/978-3-319-40578-0_7

Date deposited:

07/07/2016

Embargo release date:

30 June 2017

Reversible Computation vs. Reversibility in Petri Nets

Kamila Barylska[1], Maciej Koutny[2], Łukasz Mikulski[1], and Marcin Piątkowski[1]

[1] Faculty of Mathematics and Computer Science,
Nicolaus Copernicus University
Toruń, Chopina 12/18, Poland
{kamila.barylska,lukasz.mikulski,marcin.piatkowski}@mat.umk.pl

[2] School of Computing Science
Newcastle University
Newcastle upon Tyne, NE1 7RU, United Kingdom
maciej.koutny@newcastle.ac.uk

Abstract. Petri nets are a general formal model of concurrent systems which supports both action-based and state-based modelling and reasoning. One of important behavioural properties investigated in the context of Petri nets has been reversibility, understood as the possibility of returning to the initial marking from any reachable net marking. Thus reversibility in Petri nets is a global property. Reversible computation, on the other hand, is typically a local mechanism using which a system can undo some of the executed actions. This paper is concerned with the modelling of reversible computation within Petri nets. A key idea behind the proposed construction is to add 'reverse' versions of selected transitions. Since such a modification can severely impact on the behavior of the system, it is crucial, in particular, to be able to determine whether the modified system has a similar set of states as the original one. We first prove that the problem of establishing whether the two nets have the same reachable markings is undecidable even in the restricted case discussed in this paper. We then show that the problem of checking whether the reachability sets of the two nets cover the same markings is decidable.

Keywords: Petri net, reversibility, reversible computation, decidability

1 Introduction

Petri nets are a general formal model of concurrent systems which supports both action-based and state-based modelling and reasoning. One of important behavioural properties investigated in the context of Petri nets has been reversibility, understood as the possibility of returning to the initial marking (a global state) from any reachable marking. But it is not required that any specific transitions (global states) are used to bring the net back to the initial marking.

Reversibility in Petri nets has been investigated for years, for example, in the context of enforcing controllability in discrete event systems [18, 20, 29]. Intuitively, it is a global property which is related to the existence of home states [5, 16], i.e., those markings which can be reached from all forward reachable markings.

Unlike Petri net reversibility, reversible computation typically refers to a local mechanism using which a system can undo (the effect of) some of the already executed actions. Such an approach has been applied, in particular, to various kinds of process calculi and event structures (see, e.g., [2, 7–9, 19, 22, 23, 21]). A category theory based rendering of reversible computation with an application to Petri nets has been proposed in [10].

1.1 Previous work

A good deal of decision problems related to reversibility as well as home states and home spaces has been investigated over the past decades. These problems were usually considered within the domain of potentially infinite-state Place/Transition-net (PT-nets) and their subclasses, as most problems become trivial for finite-state net models. Typically, these problems are of one of two kinds.

In the case of the first kind of problems, one wants to establish whether a given marking (or a set of markings) satisfies a desirable property. For instance, the fundamental home state problem is concerned with establishing whether a given marking of a given PT-net is a home state. The problem was shown in [1] to be decidable, as well as its restricted version consisting in deciding whether the initial marking of a PT-net is a home state. Another example problem is that of establishing whether a linear set of markings is a home space of a given PT-net, and [11] demonstrated that such a problem is decidable. Problems of the second kind put the emphasis on the existence of a marking (or set of markings) satisfying a desirable property. For example, the fundamental home state existence problem, shown to be decidable in [4], is to establish whether there exists a home state for a given PT-net.

Although there are several positive decidability results related to reversibility, in general, the complexity of potential solutions appears to be high or difficult to establish. For example, the problem of the reversibility property is decidable but its complexity is still unknown [16], and [4] demonstrated that the problem of home state existence is at least as hard as the reachability problem [15]. This, rather pessimistic results, meant that the quest for effective algorithms, and indeed decidable problems, has for many years been carried out within special subclasses of PT-nets. Such subclasses are often defined by imposing restrictions on the structure of a net, or by assuming boundedness, with the resulting submodels of PT-nets being still relevant for a wide range of practical applications.

For example, it was shown in [6] that all live and bounded free-choice nets have home states, and the free-choice assumption cannot be changed to asymmetric choice. The home space problem is polynomial for live and bounded free-choice Petri nets [3, 12], and they also were shown to have home states [28]. Other, often progressively less restricted, net classes were considered in [3, 16, 24, 26, 27].

1.2 Our contribution

This paper is concerned with the modelling of reversible computation in Petri nets. A key idea is to add *reversed* versions of selected net transitions, each such reversed transition being obtained by simply changing the directions of adjacent arcs. The resulting reversible computations implement in a direct way what can be seen as the *undoing* of an executed action, and the simple form of such an undoing is possible thanks to the local nature of marking changes effected by net transitions.

Adding reversed transitions can greatly impact on the behavior of the system. It is therefore crucial to be able to determine whether the modified net has similar set of states as the original one. In this paper we present two key results. First, we prove that the problem of establishing whether the original net and that resulting from adding reverse transitions have the same reachable markings is undecidable even in the case of adding a single reverse. This is a strong result indicating that unless reversing of transitions is applied to restricted classes of Petri nets, such as bounded nets, controlling reversibility (so that the state space of a system does not grow) is too hard a task. We then turn to more relaxed requirement on the state space of the 'reversed' net by stipulating that what one requires is that the two nets 'cover' the same sets of markings. We then demonstrate that the problem of checking whether the reachability sets of the two nets are equivalent w.r.t. coverability is decidable.

It should be noted that focussing on coverability still has a significant application potential. For example, if all the markings covered by the original Petri net are safe on a given subset of places, then all the reachable markings of the 'reversed' net are guaranteed to be safe on this subset of places as well, provided that the nets cover the same sets of markings.

1.3 Organisation of this paper

The paper is organised as follows. In Section 2, we recall some basic definitions concerning Petri nets and their behavioural properties. Section 3 contains examples motivating our work and facilitating the understanding of the proposed approach. In Section 4, we provide the proof of undecidability of the problem of establishing whether two given nets have the same sets of reachable markings. In the Section 5, we prove that the problem of checking whether the reachability sets of two nets cover the same markings is decidable. Section 6 concludes the paper.

2 Preliminaries

The set of non-negative integers is denoted by $\mathbb{N}$. The cardinality of a set X is denoted by $|X|$, and multisets over X are members of $\mathbb{N}^X$, i.e., mappings from X to $\mathbb{N}$. If X is finite, then the multisets in $\mathbb{N}^X$ can be represented by vectors $\mathbb{N}^{|X|}$, assuming a fixed ordering of the elements of X.

The set of all multisets with componentwise addition and comparison $\leq$ is denoted by $\mathbb{N}^X$ (where $|X| \geq 1$). The componentwise subtraction is also defined if the result belongs to $\mathbb{N}^X$. One can extend the notion of $\mathbb{N}^X$ to *ω-multisets* $\mathbb{N}^X_\omega = (\mathbb{N} \cup \{\omega\})^X$, where $\omega = |\mathbb{N}|$, with the standard extensions of the addition, comparison and subtraction, assuming $\omega + n = \omega$, $\omega - n = \omega$, and $n < \omega$, for all $n \in \mathbb{N}$. The *left closures* of $y \in \mathbb{N}^X_\omega$ and $Y \subseteq \mathbb{N}^X_\omega$ are respectively defined by $\downarrow y = \{z \in \mathbb{N}^X_\omega \mid z \leq y\}$ and $\downarrow Y = \bigcup\{\downarrow y \mid y \in Y\}$. In a similar way we can define ω-vectors $\mathbb{N}^k_\omega$ as vector representations of ω-multisets.

Petri nets

A *place/transition net* (p/t-net) is a tuple $N = (P, T, W^-, W^+, M_0)$, where:

- P and T are finite disjoint sets, of *places* and *transitions*, respectively;
- $W^-, W^+ : T \to \mathbb{N}^{|P|}$ are arc weight functions; and
- $M_0 \in \mathbb{N}^{|P|}$ is the *initial marking*.

Any multiset in $\mathbb{N}^P$ is a *marking* (global state) of N, and it will be represented by a vector in $\mathbb{N}^{|P|}$, after assuming some fixed ordering of the places in P. The following terminology applies to the case of ω-markings $\mathbb{N}^P_\omega$ as well.

Petri nets admit a natural graphical representation, with nodes representing places and transitions, and annotated arcs representing the weight function. Places are indicated by circles, and transitions by boxes. For each transition $t \in T$ and place $p \in P$, $W^-(t)(p)$ is the weight of the arc from p to t, and $W^+(t)(p)$ is the weight of the arc from t to p. Arcs with zero weights are not drawn at all, and arcs with unit weights are not annotated with 1. Markings are depicted by placing tokens inside the circles.

A transition $t \in T$ is *enabled* at a marking M of N whenever $W^-(t) \leq M$. We denote this by $M[t\rangle_N$, or simply $M[t\rangle$ if N is clear from the context. If t is enabled in M, then it can be *executed*. The execution changes the current marking M to the new marking $M' = M - W^-(t) + W^+(t)$. We denote this by $M[t\rangle_N M'$, or simply $M[t\rangle M'$ if N is clear from the context.

The notions of transition enabledness and execution extend, in the usual way, to strings of transitions (computations). The empty string ε is enabled at any marking and $M[\varepsilon\rangle M$, and a string $w = tw'$ is enabled at a marking M whenever $M[t\rangle M'$ and w' is enabled at M'; moreover, $M[w\rangle M''$, where $M'[w'\rangle M''$.

If $M[w\rangle M'$, for some $w \in T^*$, then M' is *reachable from* M, and the set of all markings reachable from M is denoted by $[M\rangle_N$, or simply $[M\rangle$ if N is clear from

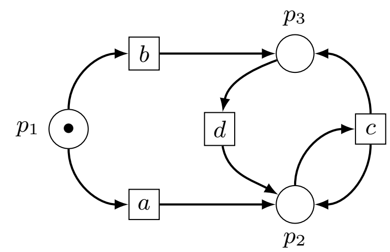

Fig. 1. A Petri net (see [25]) consisting of three places (p_1, p_2 and p_3) and four transitions (a, b, c, d).

the context. The *reachability set* of N is the set $[M_0\rangle$ of all markings reachable from the initial marking, and the markings in $[M_0\rangle$ are called *reachable* in N.

A marking M of N is a *home state* if $M \in [M'\rangle$, for every marking $M' \in [M_0\rangle$, and N is *reversible* if M_0 is a home state.

A marking $M \in \mathbb{N}^P$ *coverable* in N if there exist a reachable marking $M' \in [M_0\rangle$ such that $M \leq M'$, and $\downarrow[M_0\rangle$ is the *coverable set* of N.

A *reverse* of a transition $t \in T$ is a new transition $\bar{t}$ such that $W^-(\bar{t}) = W^+(t)$ and $W^+(\bar{t}) = W^-(t)$. To improve readability, we depict transitions of the original nets using solid lines, and the newly created reverses by dashed ones (see Figure 2).

Reachability and Coverability Graphs

Reachability graphs represent precisely the reachability sets of nets, but can be infinite, while coverability graphs are always finite, but represent precisely the coverable sets rather than reachability sets (see, e.g., [15]).

The *reachability graph* of a *p*/*t*-net $N = (P, T, W^-, W^+, M_0)$ is a directed graph $RG = ([M_0\rangle, G, M_0)$, where $[M_0\rangle$ is the set of vertices, M_0 is the initial vertex

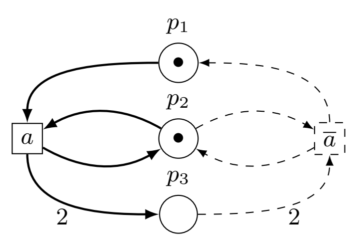

Fig. 2. A transition a and its reverse $\overline{a}$.

and $G = \{(M, t, M') \mid M \in [M_0\rangle \wedge M[t\rangle M'\}$ is the set of labelled arcs. Thus, the vertices of the reachability graph are the reachable markings of N.

In the case of a coverability graph, it is convenient to present a constructive definition based on [17].

Algorithm constructing a coverability graph

Let $N = (P, T, W^-, W^+, M_0)$ be a p/t-net. The vertices of the coverability graph constructed below are ω-vectors in $\mathbb{N}_\omega^{|P|}$.

Step 0. *Initial vertex*
We take M_0 to be the initial vertex, and set it to blue (i.e., marked). GOTO Step 1.

Step 1. *Generating new working vertices*
If there is no blue vertex then STOP. Otherwise, we take an arbitrary blue vertex M and draw from it all the arcs of the form (M, t, M'), for all $t \in T$ enabled at M (i.e., $W^-(t) \leq M$) and $M' = M - W^-(t) + W^+(t)$. If M' is not yet a vertex we add it and set to yellow (i.e., working). After drawing all such arcs we set M to grey (i.e., processed). GOTO Step 2.

Step 2. *Coverability adjustment*
If there is no yellow vertex GOTO Step 1. Otherwise, we take an arbitrary yellow vertex M and check, for all the paths from M_0 to M, whether a vertex M' such that $M' \leq M$ lies on the path and store all such vertices in $V(M)$. If $V(M) \neq \emptyset$ then every coordinate of the marking M greater than the corresponding coordinate of any marking $M' \in V(M)$ changes to ω. Finally, we set M to blue. GOTO Step 2.

The above construction always terminates, and the resulting labelled directed graph $CG = (\mathcal{M}, G^{cov}, M_0)$ is a *coverability graph* of N.

Coverability graphs are related to coverability sets (see, e.g., [13]), where a *coverability set* of N is $CS \subseteq \mathbb{N}_\omega^P$ such that the following hold:

CS1 CS covers the reachability set of N, i.e., $[M_0\rangle \subseteq\downarrow CS$; and

CS2 $[M_0\rangle$ tightly approximates all non-reachable vectors in CS, i.e., for every $M \in CS \setminus [M_0\rangle$, there is an infinite sequence of distinct markings $M_1, M_2, \ldots \in [M_0\rangle$ such that, for all $i \geq 1$:

$$M_i < M_{i+1} \quad and \quad M_{\omega/i} \leq M_i \leq M \ ,$$

where $M_{\omega/i} \in \mathbb{N}^P$ is obtained from M by replacing each ω by i.

Moreover, CS is *minimal* if no proper subset of CS is a coverability set of N.

Proposition 1. *The set of vertices of the coverability graph CG constructed above is a finite coverability set of N.*

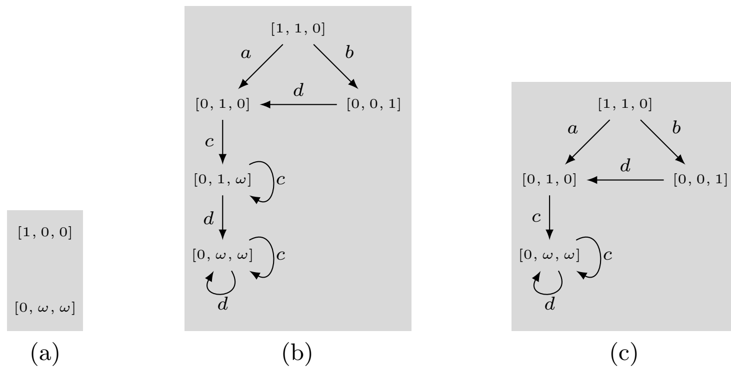

Fig. 3. The minimal coverability set (a) and two possible coverability graphs (b) and (c) of the net of Figure 1. During the generation of the graph of (b), the vertex $[0, 1, 0]$ was chosen before $[0, 0, 1]$ in the algorithm described in Section 1, while in the case of (c), the vertex $[0, 0, 1]$ was chosen before $[0, 1, 0]$.

Remark 1. Referring to [13], there exists a unique finite minimal coverability set which can be used to represent the coverable set of N, usually smaller than the set of all vertices of the coverability graph. Note that although the reachability set of N is a coverability set included in $\mathbb{N}^P$, it contains the minimal coverability set if and only if it is finite. Whenever the set of reachable markings is infinite, a finite coverability set has to use true ω-markings.

3 Motivating examples

A rather natural way of implementing the *undoing* of executed transitions is to introduce reverses of them, as shown in Figure 2. In this section, we will discuss the impact of adding reverse transitions on net behaviour.

In Figure 4, the solid lines depict a p/t-net together with its reachablility graph. Moreover, using the dashed lines, the diagram shows the reverse transition added to the original net, and the resulting enlargement of the original reachability graph. We observe that the original p/t-net was not reversible (it did not even have a home state), but the modified one is reversible and its set of reachable markings is the same as for the original net. Hence, in this case, reversing transitions 'improved' the overall net behaviour.

Figure 5 shows a p/t-net which has a home state $[0, 0, 1, 1]$. In this case, one only needs to add a reverse $\overline{b}$ of transition b to obtain a reversible net. Also, the set of reachable marking stays unchanged.

The first two examples demonstrated that adding reverse transitions can sometimes 'improve' the behaviour of the original net. In general, however, adding reverse transition changes the reachability set and also allows computations based

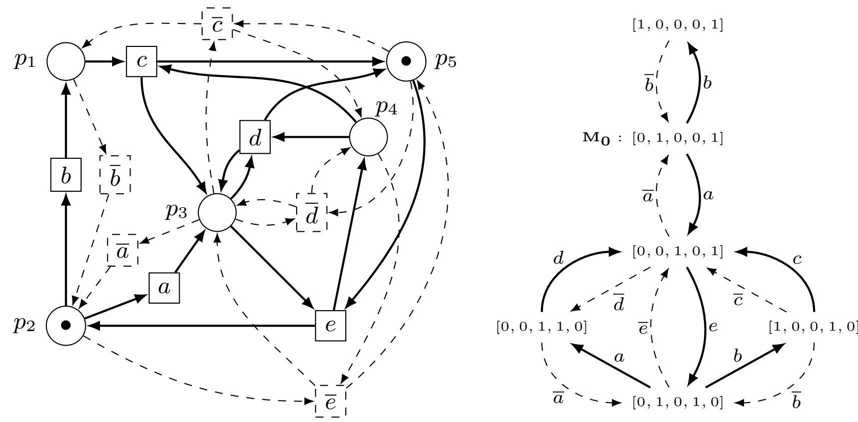

Fig. 4. A p/t-net with reverses for all transitions and its reachability graph. Reverse transitions yield reversibility.

on the original transitions which were not enabled in the original net. This may happen even if we limit ourselves to reversing only one transition.

Figure 6 shows a p/t-net with a finite set of reachable markings for which adding only one reverse $\overline{c}$ changes the reachability set to an infinite one. Note that the execution of the reverse transition $\overline{c}$ is enabled *before* the first execution of c at $[0, 1, 1]$. As a consequence, this p/t-net would model a system in which some action can be undone before it has been done, which is contrary to our intuition behind reversing a computation.

Starting with a net possessing a home state does not help either, as Figure 7 shows. The net has a home state $[0, 1, 0, 1, 1, 0]$, but, again, it is enough to add $\overline{a}$ to obtain a net with a bigger set of reachable markings.

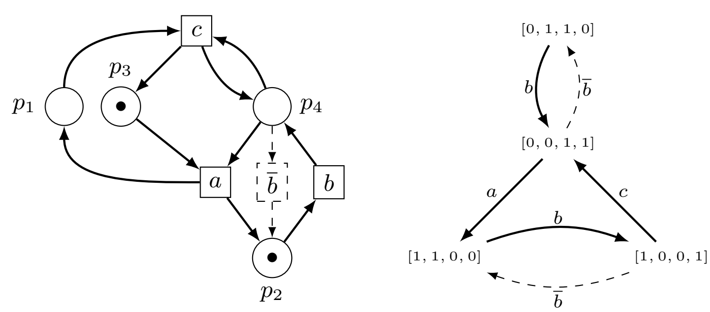

Fig. 5. A p/t-net with a single reverse transition and its reachability graph. Reverse transition yields reversibility.

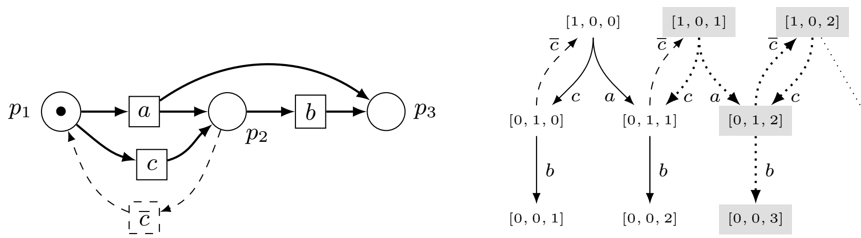

Fig. 6. A p/t-net with a single reverse transition and its reachability graph, where: the solid arcs denote arcs present in the reachability graph of the original net; the dashed arcs denote the introduced reverse of c enabled at markings reachable in the original net; and the dotted arcs represent transitions (or reverses) enabled only at markings (with gray background) which were not reachable in the original net.

The above examples suggest that it is not obvious when one can add reverses to a p/t-net without radically changing its behavior. We could also see that adding even one such transition may cause great changes in net behaviour. Thus, it is crucial to be able to decide whether a particular reverse can be added to a p/t-net without changing 'too much' its reachability set. To this end, we will discuss the decidability of the following problems involving comparisons of the state spaces of two p/t-nets.

Marking equality with single transition: MEST
: Are the reachability sets of two given p/t-nets, where the second one is obtained from the first by adding a single transition, equal?

Marking equality with single transition reverse: MESTR
: Are the reachability sets of two given p/t-nets, where the second one is obtained from the first by adding a single transition reverse, equal?

Coverable set equality: CSE
: Are the coverable sets of two given p/t-nets equal?

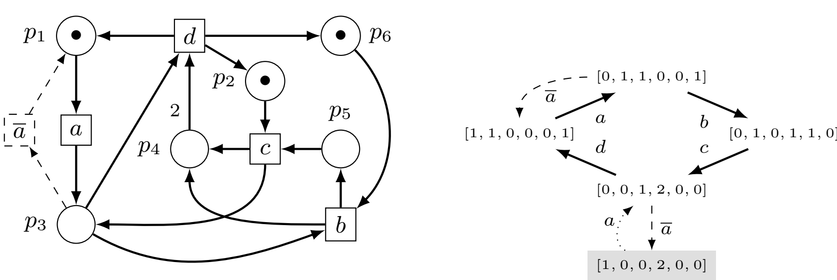

Fig. 7. A reversible p/t-net with a single reverse transition and its reachability graph. The marking with gray background is not reachable in the original net.

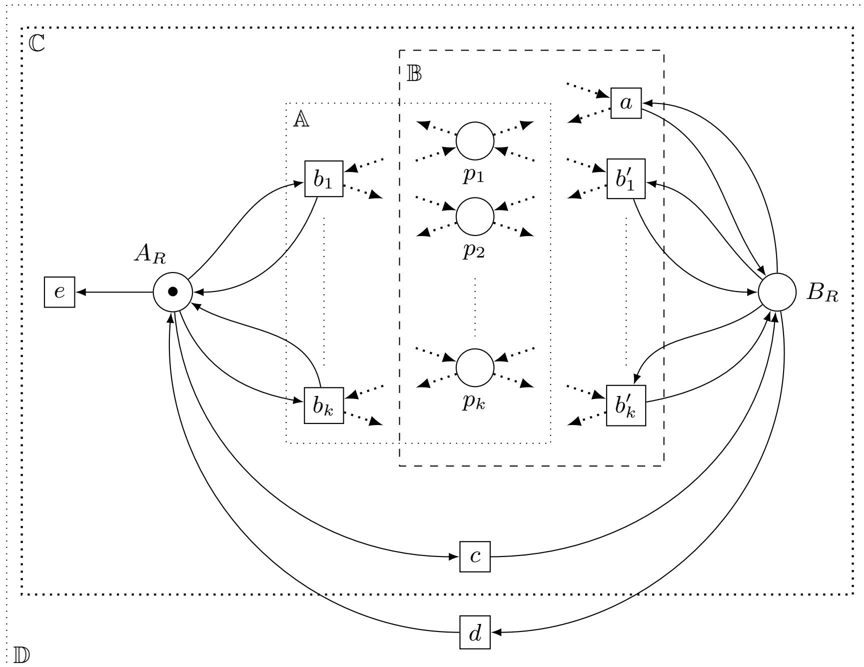

Fig. 8. The construction used in the proof of Proposition 2.

By Theorem 7.5 of [14] MEST is undecidable. In the following section, we will prove that MESTR is undecidable. Next, we will argue that CSE is decidable.

4 Undecidability of MESTR

In this section, we will show that MESTR is undecidable. The key observation is formulated as the following result.

Proposition 2. *MEST is reducible to MESTR.*

Proof. Let $\mathbb{A} = (P, T_A, W_A^-, W_A^+, M_0)$ and $\mathbb{B} = (P, T_B, W_B^-, W_B^+, M_0)$ be two p/t-nets, with the same sets of places and initial marking, and such that $T_B = \{t' \mid t \in T_A\} \uplus \{a\}$, $W_A^-(t) = W_B^-(t')$ and $W_A^+(t) = W_B^+(t')$ for every $t \in T_A$. Note that $\mathbb{B}$ can be seen as a copy or mirror[3] of $\mathbb{A}$ with an additional transition a (however, the transition sets of $\mathbb{A}$ and $\mathbb{B}$ are disjoint).

We will now describe how to construct two nets, $\mathbb{C}$ and $\mathbb{D}$, with the construction being illustrated in Figure 8.

[3] The mirror of $t_1 t_2 \dots t_k \in T_A^*$ is $t_1' t_2' \dots t_k' \in T_B^*$, and vice versa.

The net $\mathbb{C} = (P_C, T_C, W_C^-, W_C^+, M_0^C)$ is such that $P_C = P \uplus \{A_R, B_R\}$ and $T_C = T_A \uplus T_B \uplus \{c, e\}$. Moreover,

$$M_0^C(p) = \begin{cases} M_0(p) & \text{if} \quad p \in P \\ 1 & \text{if} \quad p = A_R \\ 0 & \text{if} \quad p = B_R \end{cases}$$

and the weight functions are as follows:

$$W_C^-(t)(p) = \begin{cases} W_A^-(t)(p) & \text{if} \quad p \in P & \& \quad t \in T_A \\ W_B^-(t)(p) & \text{if} \quad p \in P & \& \quad t \in T_B \\ 1 & \text{if} \quad p = A_R & \& \quad t \in T_A \\ 0 & \text{if} \quad p = B_R & \& \quad t \in T_A \\ 1 & \text{if} \quad p = B_R & \& \quad t \in T_B \\ 0 & \text{if} \quad p = A_R & \& \quad t \in T_B \\ 1 & \text{if} \quad p = A_R & \& \quad t = c \\ 0 & \text{if} \quad p \neq A_R & \& \quad t = c \\ 1 & \text{if} \quad p = A_R & \& \quad t = e \\ 0 & \text{if} \quad p \neq A_R & \& \quad t = e \end{cases}$$

and

$$W_C^+(t)(p) = \begin{cases} W_A^+(t)(p) & \text{if} \quad p \in P & \& \quad t \in T_A \\ W_B^+(t)(p) & \text{if} \quad p \in P & \& \quad t \in T_B \\ 1 & \text{if} \quad p = A_R & \& \quad t \in T_A \\ 0 & \text{if} \quad p = B_R & \& \quad t \in T_A \\ 1 & \text{if} \quad p = B_R & \& \quad t \in T_B \\ 0 & \text{if} \quad p = A_R & \& \quad t \in T_B \\ 1 & \text{if} \quad p = B_R & \& \quad t = c \\ 0 & \text{if} \quad p \neq B_R & \& \quad t = c \\ 0 & \text{if} \quad p \in P_C & \& \quad t = e \end{cases}$$

The net $\mathbb{D} = (P_C, T_D, W_D^-, W_D^+, M_0^C)$ is such that $T_D = T_C \uplus \{d\}$ and the weight functions are given by:

$$W_D^-(t)(p) = \begin{cases} W_C^-(t)(p) & \text{if} \quad p \in P & \& \quad t \in T_C \\ 1 & \text{if} \quad p = B_R & \& \quad t = d \\ 0 & \text{if} \quad p \neq B_R & \& \quad t = d \end{cases}$$

and

$$W_D^+(t)(p) = \begin{cases} W_C^+(t)(p) & \text{if} \quad p \in P & \& \quad t \in T_C \\ 1 & \text{if} \quad p = A_R & \& \quad t = d \\ 0 & \text{if} \quad p \neq A_R & \& \quad t = d \end{cases}$$

Note that in $\mathbb{D}$, transition d is the reverse of c.

In what follows, we denote a marking $M^C \in \mathbb{N}^{P_C}$ as $M_{\langle x,y \rangle}$, where $M \in \mathbb{N}^P$, $x, y \in \mathbb{N}$ and

$$M^C(p) = \begin{cases} M(p) & \text{if} \quad p \in P \\ x & \text{if} \quad p = A_R \\ y & \text{if} \quad p = B_R \end{cases}$$

The net $\mathbb{C}$ works as follows. Before the first (and only) execution of c or e we can simulate the behaviour of $\mathbb{A}$ obtaining, as a result, a marking $M^C = M_{\langle 1,0\rangle}$ such that M is any marking reachable in $\mathbb{A}$. Then there are two ways of continuing:

– After executing c we obtain $M_{\langle 0,1\rangle}$ and may proceed with the simulation of $\mathbb{B}$. Note that we can reach the same marking by executing c followed by the mirror computation in the net $\mathbb{B}$. Hence every marking reachable by some computation containing c leads to a marking $M_{\langle 0,1\rangle}$, where M is a marking reachable in $\mathbb{B}$.
– After firing e we obtain the dead marking $M_{\langle 0,0\rangle}$.

As a result, the set of reachable markings of $\mathbb{C}$ is:

$$[M_0^C\rangle_C = \{M_{\langle 0,1\rangle} \mid M \in [M_0\rangle_B\} \cup \{M_{\langle 1,0\rangle} \mid M \in [M_0\rangle_A\} \cup \{M_{\langle 0,0\rangle} \mid M \in [M_0\rangle_A\} \ .$$

The net $\mathbb{D}$ works similarly as $\mathbb{C}$. The only difference is a possible transfer of the control token from B_R to A_R using the transition d. This means that every execution in the net $\mathbb{D}$ is an alternation of executions in $\mathbb{A}$ and $\mathbb{B}$ (possibly followed by a single execution of e). As a result, from the point of view of reachable markings, we may focus only on the net $\mathbb{B}$ (starting every computation with c and ending it with d or de, if necessary). Hence, the set of reachable markings of $\mathbb{D}$ is:

$$[M_0^C\rangle_D = \{M_{\langle 0,1\rangle} \mid M \in [M_0\rangle_B\} \cup \{M_{\langle 1,0\rangle} \mid M \in [M_0\rangle_B\} \cup \{M_{\langle 0,0\rangle} \mid M \in [M_0\rangle_B\} \ .$$

We therefore conclude that $[M_0\rangle_A = [M_0\rangle_B$ if and only if $[M_0^C\rangle_C = [M_0^C\rangle_D$, which means that MEST has been reduced to MESTR. □

As a direct consequence of the above result and Theorem 7.5 of [14] we obtain

Theorem 1. *MESTR is undecidable.*

Thus verifying whether reversing a transition in a p/t-net does not change its reachability set is not a feasible problem. Clearly, for restricted classes of nets one may still look for decision procedures but, in the general case, one needs to relax the required correspondence between the state space of the original net and that resulting from reversing of some of its transitions.

5 Decidability of CSE

The construction of coverability graphs in [13] differs a bit from our approach, which is a deterministic version of Karp-Miller procedure [17]. Nevertheless, the set of labels of the coverability graph's nodes is the coverability set and so the unique minimal coverability set (see [13]) might be obtained from the set of labels of coverability graph's nodes by taking its maximal subset.

Theorem 2. *CSE is decidable.*

Proof. Let A and B be two p/t-nets with the initial markings M_0^A and M_0^B, respectively. We need to show that it is possible to effectively establish whether $\downarrow[M_0^A\rangle = \downarrow[M_0^B\rangle$. By Proposition 1, we can effectively compute finite coverability sets, CS_A and CS_B, of A and B, respectively. It then suffices to show that the following statements are equivalent:

(i) For every $M \in CS_A$, there is $M' \in CS_B$ such that $M \leq M'$.
(ii) $\downarrow[M_0^A\rangle \subseteq \downarrow[M_0^B\rangle$.

(i) $\Longrightarrow$ **(ii)** :
Suppose that $M \leq M'$ and $M' \in [M_0^A\rangle$. Then, by (CS1), there exists $M'' \in CS_A$ such that $M' \leq M''$. Thus, by (i), there exists $M''' \in CS_B$ such that $M'' \leq M'''$. If $M''' \in [M_0^B\rangle$, we get $M \in \downarrow[M_0^B\rangle$. Otherwise, by (CS2), there exists $M_i \in [M_0^B\rangle$ such that $M'' \leq M_i$. Hence, again, $M \in \downarrow[M_0^B\rangle$.

(ii) $\Longrightarrow$ **(i)** :
Suppose that $M \in CS_A$. If $M \in [M_0^A\rangle$ then, by (ii), there exists $M' \in [M_0^B\rangle$ such that $M \leq M'$. Hence, by (CS1), there exists $M'' \in CS_B$ such that $M' \leq M''$. Hence $M \leq M''$.

If $M \notin [M_0^A\rangle$ then, by (CS2), there exist distinct $M_1, M_2, \ldots \in [M_0^A\rangle$ such that, for all $i \geq 1$, $M_{\omega/i} \leq M_i \leq M_{i+1} \leq M$. Hence, by ($ii$), there exist (not necessarily distinct) $M_1', M_2', \ldots \in [M_0^B\rangle$ such that $M_{\omega/i} \leq M_i \leq M_i'$, for all $i \geq 1$. Moreover, by (CS1), there exist $M_1'', M_2'', \ldots \in CS_B$ such that $M_{\omega/i} \leq M_i' \leq M_i''$, for all $i \geq 1$. Since CS_B is finite, there exists $M' \in CS_B$ which occurs in the sequence $M_1'', M_2'', \ldots$ infinitely many times. This means that $M_{\omega/i} \leq M'$, for infinitely many i's, and so $M \leq M'$. □

Thus, in practice, we can effectively check whether the introduction of reverse transitions changes the coverable set of a p/t-net.

6 Concluding remarks

In this paper, we considered a very liberal way of reversing computation in Petri nets as it allows one to 'undo' a transition which has not yet been executed. Preventing such a behaviour would be straightforward by introducing a fresh empty 'buffer' place p_t between t and $\bar{t}$ (i.e., $W^+(t)(p_t) = W^-(\bar{t})(p_t) = 1$ $W^-(t)(p_t) = W^+(\bar{t})(p_t) = 0$). The two results we established in this paper carry over to the modified setting as in the net $\mathbb{D}$ used in the proof of Proposition 2, the executions of transitions c and d strictly alternate, starting with c.

Acknowledgements. We would like to thank the anonymous reviewers for their remarks which allowed us to improve the presentation of the paper. This work was supported by the EU COST Action IC1405, and by the Polish National Science Center (grant No. 2013/09/D/ST6/03928).

References

1. Toshiro Araki and Tadao Kasami. Decidable problems on the strong connectivity of Petri net reachability sets. *Theoretical Computer Science*, 4(1):99–119, 1977.
2. Gérard Berry and Gérard Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96(1):217–248, 1992.
3. Eike Best, Jörg Desel, and Javier Esparza. Traps characterize home states in free choice systems. *Theoretical Computer Science*, 101:161–176, 1992.
4. Eike Best and Javier Esparza. Existence of home states in Petri nets is decidable. *Information Processing Letters*, 116(6):423–427, 2016.
5. Eike Best and Uli Schlachter. Analysis of Petri nets and transition systems. In *Proceedings of 8th Interaction and Concurrency Experience (ICE'15)*, volume 189 of *EPTCS*, pages 53–67, 2015.
6. Eike Best and Klaus Voss. Free choice systems have home states. *Acta Informatica*, 21:89–100, 1984.
7. Luca Cardelli and Cosimo Laneve. Reversible structures. In François Fages, editor, *Proceedings of 9th International Computational Methods in Systems Biology (CMSB'11)*, pages 131–140. ACM, 2011.
8. Vincent Danos and Jean Krivine. Reversible communicating systems. In *Proceedings of 15th International Conference on Concurrency Theory (CONCUR'04)*, volume 3170 of *Lecture Notes in Computer Science (LNCS)*, pages 292–307. Springer-Verlag (New York), 2004.
9. Vincent Danos and Jean Krivine. Transactions in RCCS. In *Proceedings of 16th International Conference on Concurrency Theory (CONCUR'05)*, volume 3653 of *Lecture Notes in Computer Science (LNCS)*, pages 398–412. Springer-Verlag (New York), 2005.
10. Vincent Danos, Jean Krivine, and Pawel Sobocinski. General reversibility. *Electronic Notes Theoretical Computer Science*, 175(3):75–86, 2007.
11. David de Frutos Escrig and Colette Johnen. Decidability of home space property. Technical Report 503, Laboratoire de Recherche en Informatique, Université de Paris-Sud, 1989.
12. Jörg Desel and Javier Esparza. Reachability in cyclic extended free-choice systems. *Theoretical Computer Science*, 114:93–118, 1993.
13. Alain Finkel. The minimal coverability graph for Petri nets. *Lecture Notes in Computer Science*, 674:210–243, 1993.
14. Michael Hack. Decidability questions for Petri nets. Technical Report TR-161, MIT Lab. for Comp. Sci., 1976.
15. Michael Hack. Petri net languages. Technical Report TR 159, MIT Lab. for Comp. Sci., 1976.
16. Thomas Hujsa, Jean-Marc Delosme, and Alix Munier Kordon. On the reversibility of live equal-conflict Petri nets. In *Proceedings of 36th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS'15)*, volume 9115 of *Lecture Notes in Computer Science*, pages 234–253. Springer, 2015.
17. Richard Karp and Raymond Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3:147–195, 1969.
18. Danko Kezić, Nadjeljko Perić, and Ivan Petrović. An algorithm for deadlock prevention based on iterative siphon control of Petri net. *Automatika*, 47:19–30, 2006.
19. Ivan Lanese, Claudio Antares Mezzina, and Jean-Bernard Stefani. Reversing higher-order Pi. In *Proceedings of 21th International Conference on Concurrency Theory (CONCUR'10)*, volume 6269 of *Lecture Notes in Computer Science*, pages 478–493. Springer, 2010.

20. Hanife Apaydm Özkan and Aydm Aybar. A reversibility enforcement approach for Petri nets using invariants. *WSEAS Transactions on Systems*, 7:672–681, 2008.
21. Iain Phillips and Irek Ulidowski. Reversing algebraic process calculi. *Journal of Logic and Algebraic Programming*, 73(1-2):70–96, 2007.
22. Iain Phillips and Irek Ulidowski. Reversibility and asymmetric conflict in event structures. *Journal of Logic and Algebraic Methods in Programming*, 84(6):781–805, 2015.
23. Iain Phillips, Irek Ulidowski, and Shoji Yuen. A reversible process calculus and the modelling of the ERK signalling pathway. In *Proceedings of 4th Workshop on Reversible Computation (RC'12)*, volume 7581 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2012.
24. Laura Recalde, Enrique Teruel, and Manuel Silva. Modeling and analysis of sequential processes that cooperate through buffers. *IEEE Transactions on Robotics and Automation*, 14(2):267–277, 1998.
25. Wolfgang Reisig. *Petri Nets: An Introduction*, volume 4 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, Germany, 1985.
26. Enrique Teruel and Manuel Silva José Manuel Colom. Choice-free Petri nets: A model for deterministic concurrent systems with bulk services and arrivals. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 27:73–83, 1997.
27. Enrique Teruel and Manuel Silva. Liveness and home states in equal conflict systems. In *Petri Nets and Other Models of Concurrency (PETRI NETS'93)*, volume 691 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 1993.
28. Walter Vogler. Live and bounded free choice nets have home states. *Petri Net Newsletter*, 32:18–21, 1989.
29. Pengwei Wang, Zhijun Ding, and Hua Chai. An algorithm for generating home states of Petri nets. *Journal of Computational Information Systems*, 12(7):4225–4232, 2011.