ELSEVIER Theoretical Computer Science 148 (1995) 281–301

Theoretical Computer Science

Undecidability of bisimilarity for Petri nets and some related problems ☆

Petr Jančar *

Department of Computer Science, University of Ostrava, Dvořákova 7, 701 03 Ostrava, Czech Republic

Abstract

The main result shows the undecidability of (strong) bisimilarity for labelled (place/transition) Petri nets. The technique of the proof applies to the language (or trace) equivalence and the reachability set equality as well, which yields stronger versions with simpler proofs of already known results. The paper also contains two decidability results. One concerns the Petri nets which are deterministic up to bisimilarity, the other concerns semilinear bisimulations and extends the result of Christensen et al. (1993) for basic parallel processes.

1. Introduction

The topic of the paper belongs to an interesting area in the theory of parallelism and concurrency, namely to the area of decidability questions for behavioural equivalences in various classes of (models of) processes. These questions are among the first ones to ask when developing automated verification methods, for example.

There is a large amount of equivalences in the literature (cf. e.g. [28]), nevertheless some of them are felt to be more important than others. Here we are mainly interested in the equivalence called (*strong*) *bisimilarity* (or (*strong*) *bisimulation equivalence*) whose central role was recognized during the 1980s (cf. [20]).

As examples of recent contributions to the decidability-of-bisimilarity area, we could mention the results for basic process algebra (BPA) in [5] and for basic parallel processes (BPP) in [4]. It is natural to ask the relevant question also for Petri nets, one of the central models for parallelism and concurrency (cf. e.g. [23, 24] or [29]). The question had remained open for some time (mentioned explicitly e.g. in [1]), and also the decidability result for BPP, which can be viewed as a subclass of Petri nets, left the question unsolved.

Here we answer the question negatively, which yields the main result of the paper. More precisely, we show that it is undecidable whether two given labelled

☆ Supported by the Grant Agency of the Czech Republic, Grant No. 201/93/2123.

* E-mail: jancar@oudec.osu.cz

SSDI 0304-3975(95)00037-2

(place/transition) Petri nets are bisimilar or not. Bisimilarity is meant in the "classical" (interleaving) sense; some other concepts are mentioned in Section 5.

Besides the result, the technique of its proof and some consequences seem to be interesting; we discuss it in next paragraphs.

The proof is based on a well-known universal computing device, namely the register machines of Shepardson and Sturgis [25] or, equivalently, the counter machines of Minsky [21]. The universality implies the undecidability of the relevant halting problem, which is of particular interest here.

Since nonnegative counters (registers) can be regarded as places with tokens, one is naturally tempted to try to simulate a counter machine by a Petri net. But it was early recognized (cf. e.g. [23] for discussion and references) that the substantial weakness of such a "simulation" is the incapability of Petri nets to test a place (counter) for zero. A correct simulation can only be achieved by extending the power of Petri nets, e.g. by adding inhibitor arcs. In this way, we get the undecidability of the "halting problem" for such extended Petri nets unlike our "ordinary" Petri nets. In fact, Mayr [19] showed that the "halting problem" (the reachability problem, to be precise) for (ordinary) Petri nets is decidable.

Despite of the impossibility of the direct simulation, the halting problem for the counter machines can be used very naturally to show the undecidability of some other problems concerning Petri nets. In fact, we demonstrate it here on some behavioural equivalences – besides bisimilarity, we consider the language (trace) equivalence and the reachability set equality. Loosely speaking, the general strategy is the following: for a given counter machine C with given input counter values, construct two Petri nets N_1, N_2 – modifications of the "basic" net Σ_C "weakly simulating" C; the construction of N_1 and N_2 ensures that the only way to exhibit the non-equivalence is to simulate C correctly (in one of the nets) finishing in the halting state – which, of course, is possible if and only if C halts for the given input values.

Since the halting problem is undecidable even for a fixed counter machine C with 2 counters (only the input values varying; cf. [21]), we can fix the net Σ_C; in addition, its structure only allows 2 places (corresponding to the counters) to be unbounded.

In case of bisimilarity, the relevant modifications (yielding N_1, N_2) do not add unbounded places; thus we get the undecidability of the bisimilarity problem even if restricted to the labelled Petri nets with a fixed underlying (static) net and 2 unbounded places.

The same construction also yields the undecidability of the language equivalence (or trace equivalence) for labelled Petri nets with 2 unbounded places (and a fixed structure). The undecidability of the language equivalence problem is known due to Hack [9]; nevertheless his proof puts no bounds on the number of unbounded places. Later Valk and Vidal-Naquet [27] showed that labelled Petri nets with 4 and 5 unbounded places are sufficient for the undecidability.

In case of the reachability set equality, the modifications add 3 additional unbounded places; we attach a "coding subnet" to Σ_C (the subnet happens to correspond to the smallest nonsemilinear vector addition system with states [12]). Thus we get the undecidability of the reachability set equality problem even if restricted to Petri nets with 5 unbounded places. The known proofs by Rabin and Hack ([3, 10]; cf. also [23]) use Hilbert's 10th problem and Petri nets weakly computing polynomials; they do not put any bound on the number of unbounded places.

The above discussion shows that the technique of the proof for bisimilarity also provides new, technically simpler, proofs for stronger versions of some known results.

The results of this paper first appeared in the report [16]. Later Hirshfeld [11] followed the general strategy – having to invent the appropriate modifications – to show the undecidability of the language equivalence problem even for the subclass of Petri nets where each transition has one input place only; i.e., in fact, for the above mentioned BPP. Hüttel [14] then followed the approach to extend the undecidability result to some other equivalences on BPP.

The paper is completed with two decidability results concerning bisimilarity, outlined in the next paragraphs.

It is easy to see that in case of one-to-one labelled (or "unlabelled") Petri nets, bisimilarity coincides with language equivalence. Hence its decidability is clear due to reducibility of the language equivalence of these nets to the reachability problem (cf. [9, 19]). Here we show another reduction based on a "bisimulation game". To think in terms of games is often useful for understanding bisimilarity (cf. e.g. [26]); in fact, we use the game in the undecidability proof as well. Our "game" reduction (to the reachability problem) allows an easy generalization for the nets which are "deterministic up to bisimilarity". In fact, these nets correspond to $\mathscr{F}$-deterministic nets of [29]. Our "game" technique leads to a decidability proof which is short and differs substantially from the corresponding proof in [29]; it should justify including the result (Theorem 4.1) here.

Another subclass of labelled Petri nets for which the decidability of bisimilarity has been known is the above mentioned class equivalent to BPP of [4]. The proof in [4] employs a technique (suggested by Y. Hirshfeld) which is, in fact, more general – it implies decidability for the subclass where the bisimulation equivalence is a congruence w.r.t. (nonnegative vector) addition. Here the result is further extended: we show that the existence of a semilinear bisimulation is sufficient for the decidability. It is completed by the fact, known from [7], that any congruence is semilinear.

Section 2 contains basic definitions, Section 3 the undecidability results, Section 4 the decidability results. Section 5 contains some additional remarks, including e.g. the relation to the vector addition systems (with states). Some remarks also concern the notion of *weak bisimilarity*, which is touched throughout the paper as well.

2. Definitions

We begin with standard definitions and notations concerning Petri nets.

$\mathscr{N}$ denotes the set of nonnegative integers, A^* the set of finite sequences of elements of A.

A *net* is a tuple $\Sigma = (P,T,F)$ and a *labelled net* is a tuple $\Sigma = (P,T,F,L)$ where P and T are finite disjoint sets of *places* and *transitions* respectively, $F:(P\times T)\cup(T\times P)\to \mathcal{N}$ is a *flow function* (for $F(x,y) > 0$, there is an *arc* from x to y with *multiplicity* $F(x,y)$) and $L:T \to A$ is a *labelling*. We suppose a fixed (countable) set A of *actions* or *action names*; hence a labelling attaches an action name to each transition. L will also be understood in a broader sense, denoting the homomorphic extension $L:T^* \to A^*$.

A (*labelled*) *Petri net* is a tuple $N = (\Sigma, M_0)$, where Σ is a (labelled) net and M_0 is an *initial marking*, a *marking* M being a function $M:P \to \mathcal{N}$. (A marking gives the number of *tokens* for each place). A transition t is *enabled* at a marking M, denoted by $M\xrightarrow{t}_{\Sigma}$, if $M(p)\geqslant F(p,t)$ for every $p \in P$. A transition t enabled at a marking M may *fire* yielding the marking M', denoted by $M\xrightarrow{t}_{\Sigma}M'$, where $M'(p) = M(p)-F(p,t)+F(t,p)$ for all $p \in P$. For any $a \in A$, by $M\xrightarrow{a}_{\Sigma}$ ($M\xrightarrow{a}_{\Sigma} M'$) we mean that $M\xrightarrow{t}_{\Sigma}$ ($M\xrightarrow{t}_{\Sigma} M'$) for some t with $L(t) = a$. In the natural way, the definitions can be extended for finite sequences of transitions $\sigma \in T^*$ and finite sequences of actions $w \in A^*$.

The *reachability set* of a Petri net $N = (\Sigma, M_0)$ is defined as $\mathscr{R}(N) = \{M \mid M_0\xrightarrow{\sigma}_{\Sigma} M$ for some $\sigma\in T^*\}$. A *place* $p \in P$ is *unbounded* if for any $k \in \mathcal{N}$ there is $M \in \mathscr{R}(N)$ s.t. $M(p) > k$.

The *language*, or the *set of traces*, of a labelled Petri net N is defined as $\mathscr{L}(N) = \{w\in A^* \mid M_0\xrightarrow{w}_{\Sigma}\}$. Two labelled Petri nets N_1, N_2 are *language equivalent* if $\mathscr{L}(N_1) = \mathscr{L}(N_2)$.

Now we give a standard definition of (strong) bisimulation and bisimilarity.

Given two labelled nets $\Sigma_1 = (P_1,T_1,F_1,L_1)$, $\Sigma_2 = (P_2,T_2,F_2,L_2)$, a binary relation $R \subseteq \mathcal{N}^{P_1} \times \mathcal{N}^{P_2}$ is a *bisimulation* if for all $(M_1,M_2) \in R$:

(1) for each $a\in A$ and M_1' s.t. $M_1\xrightarrow{a}_{\Sigma_1} M_1'$ there is M_2' s.t. $M_2\xrightarrow{a}_{\Sigma_2} M_2'$ and $(M_1',M_2') \in R$, and conversely

(2) for each $a\in A$ and M_2' s.t. $M_2\xrightarrow{a}_{\Sigma_2} M_2'$ there is M_1' s.t. $M_1\xrightarrow{a}_{\Sigma_1} M_1'$ and $(M_1',M_2') \in R$.

Two labelled Petri nets N_1,N_2 are *bisimilar* if there is a bisimulation relating their initial markings.

As mentioned in the Introduction, it is often useful to think in terms of a game (Stirling illustrates it e.g. in [26]). We recall a definition which will help us in informal arguing (for brevity, we use "he" referring to a player, instead of "he or she").

Bisimulation game

1. *Prerequisities*
   There are two players, Player 1 and Player 2, and a pair of labelled Petri nets N_1,N_2 (as the "playboard").
2. *Rules*
   Player 1 chooses one of the nets and fires an enabled transition (changing the marking appropriately); let us denote its label by a. Then Player 2 responds by firing a transition with the same label a in *the other* net (if it is possible). Again, Player 1 chooses one of the nets ..., Player 2 responds etc.

3. *Result*
   The player who has no possible move (being his turn) loses; the other player wins. The case of an infinite run of the game is considered to be successful for Player 2 (he defends successfully).

The relation of the definition and the game is expressed in the next proposition, which is almost trivial.

Proposition 2.1. *Player 1 has a winning strategy in the bisimulation game iff* N_1, N_2 *are not bisimilar; in other words, Player 2 has a defending strategy iff* N_1, N_2 *are bisimilar.*

Proof. If there is a relevant bisimulation R, Player 2 is always able to respond in such a way that the current markings are related by R; hence he has a defending strategy. If Player 2 (has and) uses a defending strategy then all pairs of markings which can appear during the game after a move of Player 2 yield, in fact, a bisimulation. □

Thinking in terms of the game, the following useful proposition is immediately clear.

Proposition 2.2. *If two labelled Petri nets* N_1, N_2 *are bisimilar then they are language equivalent, i.e.* $\mathscr{L}(N_1) = \mathscr{L}(N_2)$.

Since the notion of *weak bisimilarity* is also touched in the paper, we outline the respective definitions briefly. In this case, it is assumed that the set A of actions contains a special – unobservable – action τ. Now, for any $a \in A$, $M \stackrel{a}{\Longrightarrow}_{\Sigma} M'$ means that $M \stackrel{\sigma}{\longrightarrow}_{\Sigma} M'$ for some sequence σ of transitions, one of them being labelled by a, the others by τ; in case $a = \tau$, the sequence can be empty (i.e. $M \stackrel{\tau}{\Longrightarrow}_{\Sigma} M$ for all markings M). A *weak bisimulation* is defined in the same way as the (strong) bisimulation but the relation $\stackrel{a}{\longrightarrow}_{\Sigma}$ is replaced with $\stackrel{a}{\Longrightarrow}_{\Sigma}$. Two labelled Petri nets N_1, N_2 are *weakly bisimilar* if there is a weak bisimulation relating their initial markings. The corresponding adjustments of the bisimulation game and Proposition 2.1. are straightforward.

3. Undecidability results

As follows from the Introduction, we use reductions of the halting problem for the counter machines to our problems considered. Let us recall a standard definition.

Definition 3.1. *A counter machine C* with nonnegative counters $c_1, c_2, \ldots, c_m$ is a program

$$1 : COMM_1;\ 2 : COMM_2;\ \ldots\ ;\ n : COMM_n$$

where $COMM_n$ is a *HALT*-command and $COMM_i$ $(i = 1, 2, \ldots, n-1)$ are commands of the following two types (assuming $1 \leqslant k, k_1, k_2 \leqslant n, 1 \leqslant j \leqslant m$)

(1) $c_j := c_j + 1;$ *goto* k

(2) *if* $c_j = 0$ *then goto* k_1 *else* $(c_j := c_j - 1;$ *goto* $k_2)$
The set *BS* of *branching states* is defined as $BS = \{i \mid COMM_i$ is of the type $2\}$.

In the next definition we describe a construction, which could be considered as a "first attempt" to simulate a counter machine by a Petri net. We also define an auxiliary notion of a "definitely cheating" transition. In the construction, as well as in the further ones, we describe a successive adding of new places, transitions and arcs. By adding an arc (x, y) we mean putting $F(x, y) = 1$ unless otherwise stated; at the pairs (x, y) for which the arc is not added we (implicitly) put $F(x, y) = 0$.

Definition 3.2. Let C be a counter machine in the above notation. By the *basic net* Σ_C we mean the net constructed as follows.

Construction of Σ_C

1. Let $c_1, c_2, \ldots, c_m$ (the counter part) and $s_1, s_2, \ldots, s_n$ (the state part) be places of Σ_C .
2. For $i = 1, 2, \ldots, n-1$ add new transitions and arcs depending on the type of $COMM_i$:

Case 1: $COMM_i$ is $c_j := c_j + 1;$ *goto* k :
Add t_i with (s_i, t_i), (t_i, c_j), (t_i, s_k) (cf. Fig. 1(a))

Case 2: $COMM_i$ is *if* $c_j = 0$ *then goto* k_1 *else* $(c_j := c_j - 1;$ *goto* $k_2)$:
add t_i^Z (Z for zero) with (s_i, t_i^Z), (t_i^Z, s_{k_1}), and
t_i^{NZ} (NZ for nonzero) with (s_i, t_i^{NZ}), (c_j, t_i^{NZ}), (t_i^{NZ}, s_{k_2}) (cf. Fig. 1(b)).

Adding a dc-*transition* (dc for "definitely cheating") to Σ_C for some $i \in BS$ means adding a new transition t with (s_i, t), $(c_j, t), (t, c_j)$, (t, s_{k_1}), j, k_1 taken from $COMM_i$ (cf. Fig. 1(c)).

Notice that, for any input counter values $x_1, x_2, \ldots, x_m$ if we put 1 token in the place s_1 of Σ_C, $x_1, x_2, \ldots, x_m$ tokens in the places $c_1, c_2, \ldots, c_m$ respectively and 0 tokens

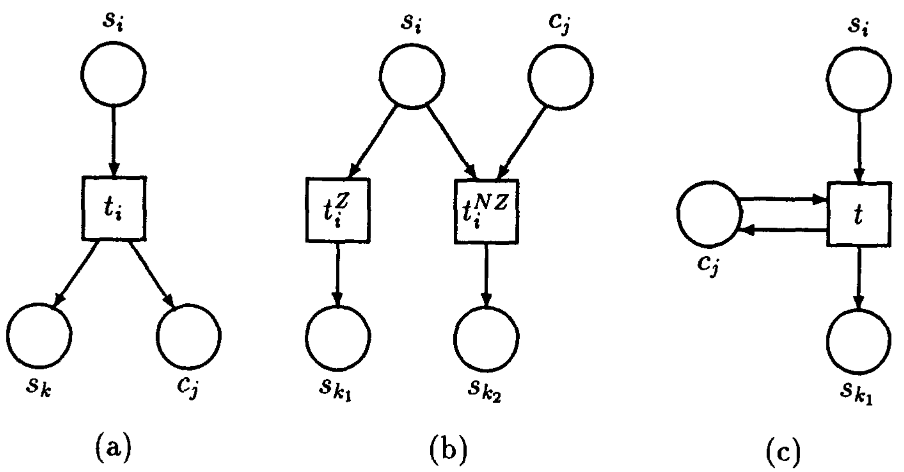


Fig. 1. Construction of Σ_C and dc-transitions.

elsewhere then the arising Petri net can "simulate" C in a natural way but ("only") the transitions t_i^Z can *cheat*, i.e. fire although the relevant c_j is not 0. Also notice that a dc-transition t has the same effect as the relevant t_i^Z but firing it always means cheating.

The construction of Σ_C applies to any counter machine. Nevertheless it will suffice to consider a fixed counter machine C with two counters c_1, c_2 for which the halting problem is undecidable (it is undecidable for given input values x_1, x_2 of c_1, c_2 whether C halts or not); the existence of such a ("universal") C is well-known (cf. [21]). We refer to such a (fixed) machine C in the proofs of the next theorems.

Theorem 3.3. *Bisimilarity as well as language equivalence are undecidable for labelled Petri nets, even if restricted to the subclass with a fixed underlying net and 2 unbounded places.*

Proof. We follow the general strategy mentioned in the Introduction. For the fixed (universal) counter machine C and two input values x_1, x_2, we will construct two labelled Petri nets N_1, N_2 (modifications of Σ_C) such that the following conditions are equivalent

(a) C does not halt for the given inputs x_1, x_2,
(b) N_1, N_2 are bisimilar,
(c) $\mathscr{L}(N_1) = \mathscr{L}(N_2)$,
(d) $\mathscr{L}(N_1) \subseteq \mathscr{L}(N_2)$.

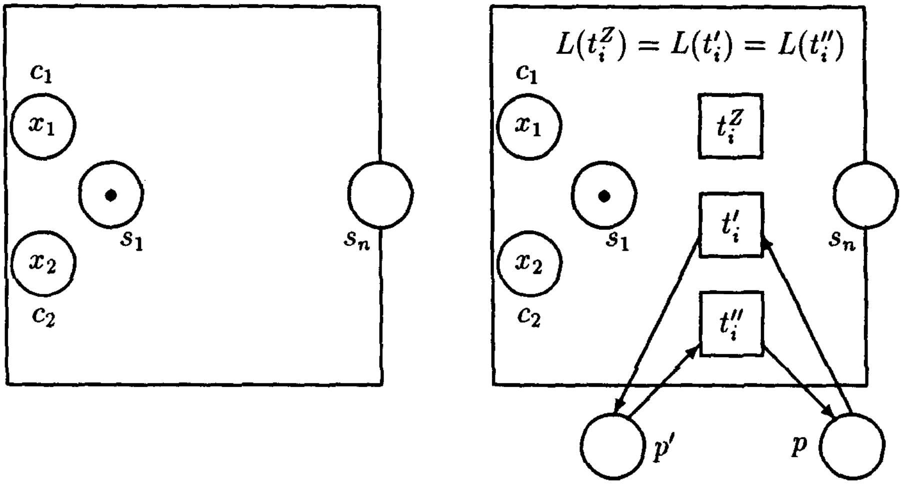
It will immediately imply the theorem. Condition (d) (directly implying the undecidability of the language containment problem) could be omitted; nevertheless including it does not add any work and can be useful for understanding.

Before describing the construction, notice that (b) implies (c) (cf. Proposition 2.2.) and (c) implies (d). Therefore it will suffice to show that (a) implies (b) and that (d) implies (a) (i.e. non-(a) implies non-(d)).

Construction of N_1, N_2

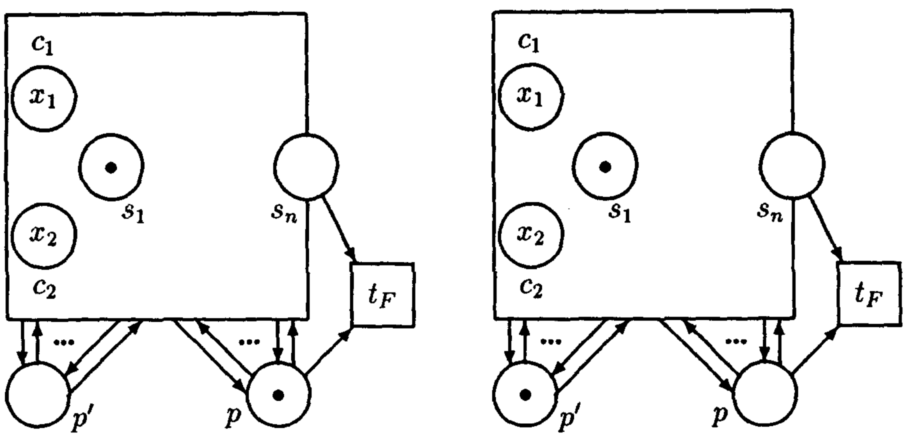
1. As a basis, take Σ_C, put 1 token in s_1, x_1 tokens in c_1, x_2 tokens in c_2 and 0 elsewhere (as sketched in Fig. 2(a)).
2. Add new places p, p' and take any one-to-one labelling L of transitions.
3. For each $i \in BS$, add two dc-transitions t_i', t_i'' (with the relevant arcs), the additional arcs (p, t_i'), (t_i', p'), (p', t_i''), (t_i'', p), and put $L(t_i') = L(t_i'') = L(t_i^Z)$ (Fig. 2(b) sketches it for one $i \in BS$).
4. Add a new transition t_F with a new label (extending the labelling L) and the arcs (s_n, t_F), (p, t_F).
5. Now take two copies of the arising net, as indicated in Fig. 3.
   In one copy put 1 token in p and 0 in p', which yields N_1.
   In the other copy put 1 token in p' and 0 in p, which yields N_2.

It is clear that only c_1, c_2 are (possibly) unbounded. Also notice that, in any reachable marking, p and p' together hold at most one token.



(a) (b)

Fig. 2. Auxiliary constructions for bisimilarity problem.



(a) (b)

Fig. 3. N_1 and N_2 for bisimilarity problem.

Now we show the promised implications "non-(a) implies non-(d)" and "(a) implies (b)". In doing this we refer to the bisimulation game described in Section 2.

If C halts (for inputs x_1,x_2): Then there is a simple winning strategy for Player 1 – to fire, step-by-step, a fixed sequence σ in N_1 (independent of Player 2's responses); it implies not only nonbisimilarity of N_1 and N_2 but also $\mathscr{L}(N_1) \not\subseteq \mathscr{L}(N_2)$ (because $L(\sigma)$ belongs to $\mathscr{L}(N_1)$ and not to $\mathscr{L}(N_2)$). The mentioned σ is the transition sequence which simulates C correctly (there is no cheating) and is finished by t_F. In each move, Player 2 has to answer in N_2 by firing a transition with the same label. Since Player 1 does not fire any dc-transition and any firing of t_i^Z is correct (the relevant counter-place is empty), Player 2 has, in fact, no choice and has to follow Player 1 by firing the same transition sequence in N_2. But it is clear that he can not finish by t_F (not having a token in p).

If C does not halt (for inputs x_1,x_2): We show the bisimilarity of N_1, N_2 by describing a defending strategy for Player 2:

(1) if Player 1 makes a correct move in one net then fire the same transition in the other net (anyway, there is no choice),

(2) if Player 1 makes a cheating move in one net, i.e. fires (for some $i \in BS$) a dc-transition t_i' or t_i'', or t_i^Z when the relevant c_j is not empty, then fire *the* transition out of the set $\{t_i^Z, t_i', t_i''\}$ in the other net which ensures the same markings on places p, p' (and hence on all places !) in both nets; in other words, fire *the* enabled dc-transition out of $\{t_i', t_i''\}$ if the current markings differ w.r.t. p, p', and fire t_i^Z otherwise.

It is clear that Player 1 cannot win by firing correct moves only (since C does not halt, he cannot use t_F); but after a first cheating move, Player 2 reaches the same markings in both nets – a clearly bisimilar situation!

Of course, we could proceed more formally and extract a concrete bisimulation from the above considerations. An example of a bisimulation (relating the initial markings) is the union $\mathscr{D} \cup \mathscr{M}$ where $\mathscr{D}$ is the diagonal (the set of all pairs (M,M)) and $\mathscr{M}$ is the set of all pairs (M',M'') where M' and M'' are reachable by correct sequences of the same lenghts in N_1 and N_2 respectively (it also means $M'(p) = 1, M'(p') = 0$ and $M''(p) = 0, M''(p') = 1$). □

Remark 3.4. Considering language equivalence only, we could use a simpler, "non-symmetric", construction: N_1 without p' and dc-transitions, N_2 with only one set of dc-transitions moving the token from p' to p.

Now we show an auxiliary Petri net (a modification of Σ_C enriched by a "coding subnet") which serves as a basis for the proof of Theorem 3.7.

Definition 3.5. Let C be the fixed counter machine and x_1, x_2 some input values. By N_{x_1,x_2} we mean the Petri net constructed as follows (cf. Fig. 4).

Construction of N_{x_1,x_2}

1. Take the basic net Σ_C, put 1 token in s_1,x_1 tokens in c_1,x_2 tokens in c_2 and 0 elsewhere.
2. Add a dc-transition t_i' (with the relevant arcs) for each $i \in BS$.
3. Add places $COD, HELP, SC$ (step counter) and r_1, r_2; put 1 token in r_1,0 in the others.
4. Add arcs $(r_1,t),(t,r_2),(t,SC)$ for each (so far constructed) transition t and $(t_i^{NZ}$, $COD)$ for each t_i^{NZ}; we call the so far constructed transitions *counted* (in SC).
5. Add transitions u_1, u_2, u_3 and arcs (COD,u_1), (r_2,u_1), (u_1,r_2), $(u_1,HELP)$ with $F(u_1,HELP) = 2$, (r_2,u_2), (u_2,r_1), $(HELP,u_3)$, (r_1,u_3), (u_3,r_1), (u_3,COD).
6. Thus the construction of N_{x_1,x_2} is completed.

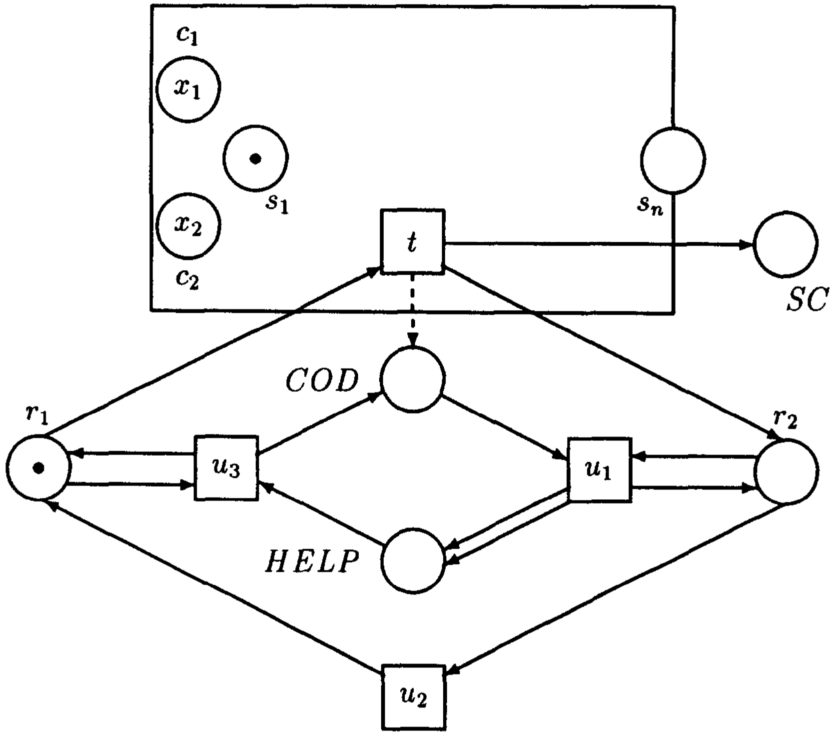


Fig. 4. Construction of N_{x_1,x_2}.

The "computation" of N_{x_1,x_2} runs in certain *cycles*; each cycle consists of firing a (*non-u_i*) transition t *counted* in *SC* and of firing a sequence $\sigma_u = u_1u_1\ldots u_1u_2u_3u_3\ldots u_3$ (for some finite numbers of u_1 and u_3). Firing t moves the token from r_1 to r_2, adds a token to *SC*; if $t = t_i^{\mathrm{NZ}}$ for some i, a token is also added to *COD*. The sequence σ_u moves the token from r_2 back to r_1, (possibly) changing (the number of tokens in) *COD*. Notice that, supposing *HELP* empty, the maximal possible increasing of *COD* in one cycle can be expressed as $COD := 2 \cdot COD$ or $COD := 2(COD + 1)$ (for t_i^{NZ}).

We will need the following fact.

Lemma 3.6. *Suppose that the computation of C on x_1, x_2 takes k steps at least. Then the following sequence σ is enabled in N_{x_1,x_2}: $\sigma = t^1\sigma_u^1t^2\sigma_u^2\ldots t^k\sigma_u^k$ where firing each (counted) t^i is correct (non-cheating) and each σ_u^i ($i = 1,2,\ldots,k$) causes the maximal possible increasing of COD; let M denote the marking reached by firing σ. In addition, for any other enabled sequence σ' with k (occurrences of) counted transitions and the respective marking M', we have $M'(COD) < M(COD)$.*

Proof. It is clear that σ is enabled, the maximality of *COD* in that case is the point.

First note the easy fact that it suffices only to consider the sequences $\sigma' = z^1\delta_u^1z^2\delta_u^2\ldots z^k\delta_u^k$ where $z^1, z^2, \ldots, z^k$ are the (only) counted transitions in σ' and each δ_u^i ($i =$

$1,2,\ldots,k$) causes the maximal possible increasing of COD; if some δ_u^i did not mean the maximal increasing then the reached marking would be smaller on COD (later δ_u^j, $j > i$, could not restore it).

Having such a σ', and the reached marking M', it can be easily verified by induction on k, that $M'(COD) = b_1 b_2 \ldots b_k b_{k+1}$ in binary where $b_{k+1} = 0$ and, for $j = 1,2,\ldots,k$, $b_j = 1$ if $z^j = t_i^{\mathrm{NZ}}$ for some i and $b_j = 0$ otherwise.

If $\sigma' \neq \sigma$ (and σ' is enabled in N_{x_1,x_2}) then there is j $(1 \leqslant j \leqslant k)$ s.t. $t^1 = z^1$, $t^2 = z^2,\ldots,t^{j-1} = z^{j-1}$ and t^j is an "NZ-transition" (i.e. t^j is t_i^{NZ} for some i) and z^j is a "non-NZ-transition" (i.e. $z^j = t_i^{\mathrm{Z}}$ or $z^j = t_i'$). Then the inequality $M'(COD) < M(COD)$ is clear from the binary expressions. □

In the next theorem, we mention the containment problem for reachability sets explicitly although its undecidability follows from the undecidability of the equality problem. The reason is similar to the language containment problem mentioned in the proof of Theorem 3.3; in addition, recall that the proof for the containment problem [3] preceded the proof for the equality problem (cf. [10]).

Theorem 3.7. *The containment and the equality problems for reachability sets of Petri nets are undecidable, even if restricted to the subclass of Petri nets with 5 unbounded places (and with one of two fixed underlying nets).*

Proof. For the fixed (universal) counter machine C and two input values x_1,x_2, we will construct two Petri nets N_1, N_2 (modifications of N_{x_1,x_2}) such that the following conditions are equivalent

(a) C does not halt (for the given inputs x_1,x_2)

(b) $\mathscr{R}(N_1) = \mathscr{R}(N_2)$

(c) $\mathscr{R}(N_1) \subseteq \mathscr{R}(N_2)$

Since (b) implies (c), it will suffice to show that non-(a) implies non-(c) and that (a) implies (b).

Construction of N_1, N_2

1. Take N_{x_1,x_2} as a basis and add a place p with 1 token and a place p' with 0 tokens. Then add the arcs (p,t_i'), (t_i',p') for each dc-transition ($i \in BS$); Fig. 5 illustrates it for one of these transitions.
2. Add a transition t_a with the arc (s_n,t_a). The arising net is the desired N_2.
3. N_1 arises from N_2 by adding an additional transition t_b with the arcs (s_n,t_b), (p,t_b), (t_b,p'). Fig. 6 shows N_1; removing the transition t_b (in the dashed box) with the adjacent arcs, we get N_2.

Trivially $\mathscr{R}(N_2) \subseteq \mathscr{R}(N_1)$. Notice that at most one dc-transition can fire (in any of N_1, N_2), moving the token from p to p'. Also notice that only places $c_1,c_2,COD,HELP,SC$ are (possibly) unbounded.

Now we show the promised implications "non-(a) implies non-(c)" and "(a) implies (b)".

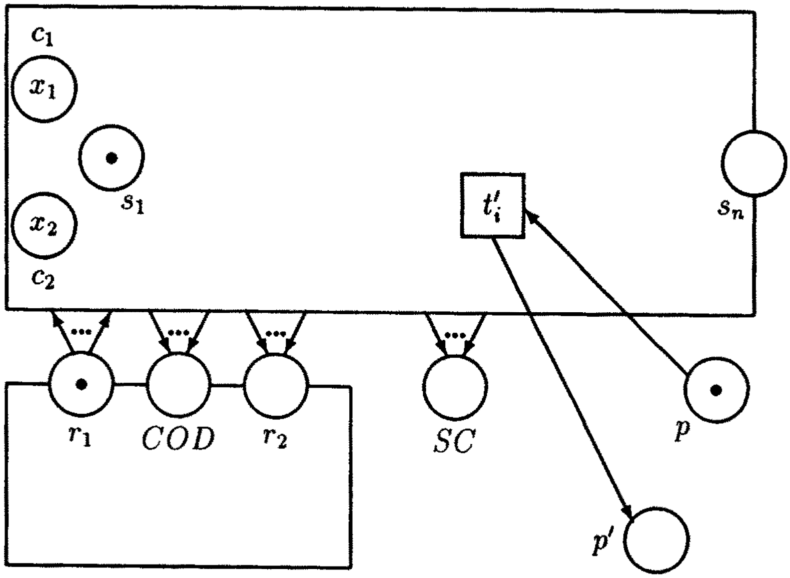


Fig. 5. Auxiliary construction for equality problem.

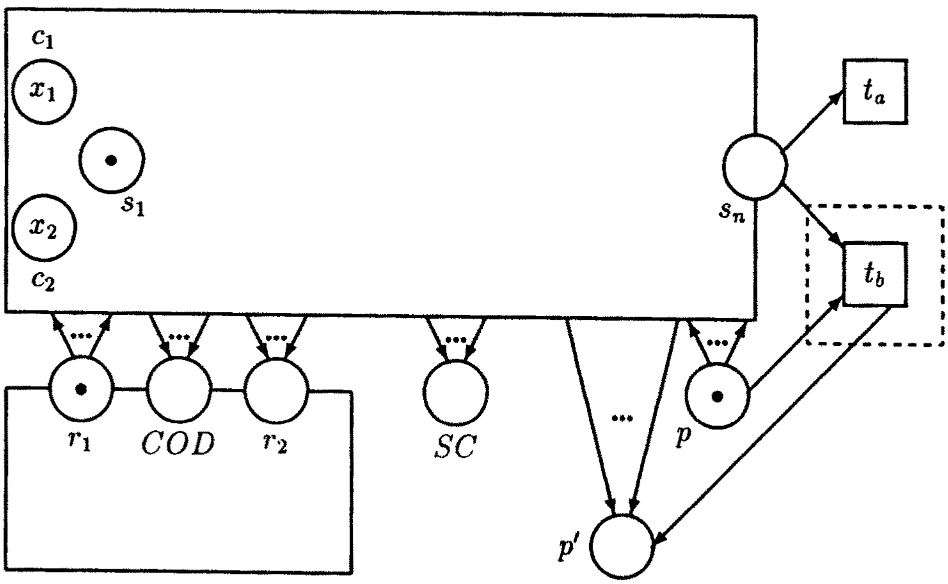


Fig. 6. N_1 and N_2 for equality problem.

If C halts (for inputs x_1,x_2*)*: Consider firing the correct (noncheating) $\sigma = t^1\sigma_u^1 t^2\sigma_u^2 \ldots t^k\sigma_u^k t_b$ in N_1 which finishes by t_b and all the intermediate changes of *COD* are maximal; M will denote the reached marking.

Let us ask if there is a sequence σ' fireable in N_2 which would also reach the marking M. Since $M(SC) = k$, σ' has to contain exactly k counted transitions. As σ'

should reach M on COD as well, its prefix has to be $t^1\sigma_u^1 t^2\sigma_u^2 \ldots t^k\sigma_u^k$ due to Lemma 3.6. Then only t_a can follow (in fact, t_a can be inserted anywhere into σ_u^k) and the token is left in p ($M'(p) = 1$ for the reached marking M') – whereas $M(p) = 0$. Hence there is no such σ' and it means $\mathscr{R}(N_1) \not\subseteq \mathscr{R}(N_2)$.

If C does not halt (for inputs x_1, x_2): N_1 and N_2 only differ in the transition t_b. This transition can fire (in N_1) once at most and no counted transition can fire after it (only u_i-transitions might follow). Consider any fireable sequence σ in N_1 which contains t_b. It is clear that σ contains no dc-transition t_i' (it would disable t_b removing the token from p) and that a token has been put in the $HALT$ place s_n before firing t_b – therefore at least one firing of t_i^Z was cheating. The same marking is reached in N_2 by firing a sequence σ' which arises from σ by replacing an occurrence of a (cheating) t_i^Z with t_i' and the occurrence of t_b with t_a. Hence $\mathscr{R}(N_1) = \mathscr{R}(N_2)$. □

Remark 3.8. In Section 5 we also mention similar problems for the vector addition systems with states (VASSs). Having VASSs in mind, the restrictions of reachability sets to unbounded places are important rather than the reachability sets themselves. In the above construction these restrictions (of $\mathscr{R}(N_1)$, $\mathscr{R}(N_2)$ to the domain $\{c_1, c_2, COD, HELP, SC\}$) are equal also in the case that C halts; hence the proof does not show directly that the equality problem for 5-dim VASS is undecidable. Nevertheless, the proof can be easily modified to show it. E.g. we can cancel p', directing its input arcs to COD, and increase the multiplicity of the arc $(u_1, HELP)$ to 3 and the multiplicity of all arcs (t_i^{NZ}, COD) to 2 – to maintain the superiority of t_i^{NZ} over t_i'.

4. Decidability results

This section shows two subclasses of the class of labelled Petri nets for which bisimilarity is decidable.

The first subsection shows a natural possibility to use the bisimulation game for obtaining a decidability result. The relevant subclass – Petri nets which are deterministic up to bisimilarity – corresponds to $\mathscr{F}$-deterministic nets of [29] (cf. Section 5) and therefore the result is not new, in fact. It is its proof technique which should justify (as the author hopes) its including here.

As already mentioned, the result in [4] for BPP yields another subclass of Petri nets for which bisimilarity is decidable. We show in the second subsection that the result (and the corresponding subclass) can be extended.

4.1. Deterministic nets

Let us begin with considering one-to-one labelled (or “unlabelled”) Petri nets. Such Petri nets N_1, N_2 are bisimilar iff $\mathscr{L}(N_1) = \mathscr{L}(N_2)$; it can be easily verified when thinking in terms of the bisimulation game (described in Section 2). The language equivalence problem in this case is known to be reducible to the reachability problem

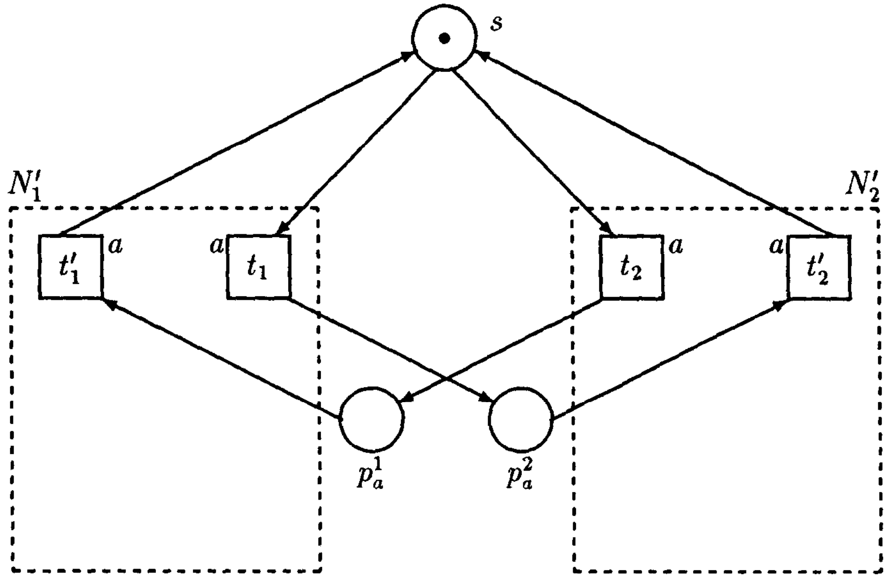


Fig. 7. Game net N.

(cf. [9]), which is known to be decidable (cf. [19]). (The *reachability problem* is to decide for a given Petri net N and a marking M whether $M \in \mathscr{R}(N)$.)

Now we use our bisimulation game to show another reduction, which allows a straightforward generalization. It is convenient to model the game within one Petri net – the *game net N*.

Construction of the game net N (cf. Fig. 7)

1. Suppose that two labelled Petri nets $N_1 = (P_1, T_1, F_1, L_1, M_{01})$, $N_2 = (P_2, T_2, F_2, L_2, M_{02})$ are given.
   For each transition $t \in T_1$, add a *duplicate* transition t'; the arising net will be denoted by $N_1' = (P_1, T_1', F_1', L_1', M_{01})$; i.e. $F_1'(p,t') = F_1(p,t)$, $F_1'(t',p) = F_1(t,p), L_1'(t') = L_1(t)$ for all $t \in T_1$, $p \in P_1$.
   Similarly add duplicate transitions to N_2 yielding the net N_2'.
2. Take the union of N_1', N_2' (simply put N_1', N_2' beside each other; we suppose $P_1 \cap P_2 = \emptyset$ and $T_1' \cap T_2' = \emptyset$).
   To the arising net, add a new place s with 1 token and for any $a \in A$ add places p_a^1, p_a^2 with 0 tokens.
3. For any $t_1 \in T_1$: add (s,t_1), (t_1,p_a^2), (p_a^1,t_1'), (t_1',s) where $a = L_1(t_1)$ (t_1' being the duplicate of t_1).
   For any $t_2 \in T_2$: add (s,t_2), (t_2,p_a^1), (p_a^2,t_2'), (t_2',s), where $a = L_2(t_2)$.
   Thus the construction of N is completed.

The game in the net N, equivalent to the bisimulation game, can be given simpler rules: Player 1 fires an enabled transition (of N), then Player 2 fires an enabled transition (of N), Player 1 fires,..., etc. Player 1's winning corresponds to any situation (marking) where a token is in some p_a^i and no transition is enabled.

Reachability of such a marking is decidable – it is a technical routine to show a reduction to the reachability problem; we give the relevant lemma at the end of this subsection.

In general, reachability of a winning situation does not mean that Player 1 has a winning strategy; Player 2 could possibly have avoided the situtation by choosing more clever answers before. But if the labelling is one-to-one, he could not have answered in a different way; hence the reachability of a winning situation really means the existence of a winning strategy.

It is clear that the same argument applies if the nets are *deterministic*, i.e. no reachable marking enables two different transitions with the same label. We can even allow the nets to be *deterministic up to bisimilarity* – in such a net, different transitions with the same label can be enabled but their firings lead to bisimilar results. It even suffices when *one* of the nets is deterministic up to bisimilarity; the winning strategy is then to follow the relevant path in *the other* net, giving, in fact, no choice to Player 2. Thus we have proved the following theorem.

Theorem 4.1. *Bisimilarity is decidable for two labelled Petri nets, supposing one of them is deterministic up to bisimilarity (hence if one of them is deterministic, hence if one of them is one-to-one labelled).*

Remark 4.2. It is not difficult to modify the construction of the game net for the case of weak bisimilarity; in any move, each Player fires a sequence of transitions with at most one non-τ-label. Player 1's winning situation would correspond to a reachable marking where a token is in some p_a^i, $a \neq \tau$, and no τ-sequence of transitions could enable a transition labelled with a. By standard means (using e.g. the reachability tree, cf. [23]), it can be shown that the (finite) set of all minimal markings where a τ-sequence could enable an a-transition is effectively constructible. Hence the reachability of a Player 1's winning marking is decidable (the marking can be described in the extended L_Σ defined below). Thus "bisimilarity" can be replaced with "weak bisimilarity" in Theorem 4.1. In Section 5 we mention connections with the results of [29] concerning $\mathscr{F}$-deterministic Petri nets.

The decidability proof for the above mentioned problem(s) uses the decidability of the reachability problem. The next lemma completes this by showing that they are at least as hard as that problem.

Lemma 4.3. *The reachability problem is PTIME-reducible to the bisimilarity problem for one-to-one labelled Petri nets.*

Proof. It is well-known (cf. e.g. [9] or [23]) that the reachability problem is recursively (in fact, P-TIME) equivalent to the problem *SPZRP* of finding out if a given (single) place can be empty (with 0 tokens) in some reachable marking. We will reduce this problem to the bisimilarity problem.

Consider the following construction for an instance of *SPZRP*, i.e. a Petri net N and a place p.

Construction of N_1, N_2

1. To N, add a transition t, a place r with 1 token and $(r,t),(t,r)$.
2. Take any one-to-one labelling of transitions; denote the arising net by N_1.
3. Construct N_2 from N_1 by adding the arcs $(p,t),(t,p)$.

If $M(p) \geqslant 1$ for all $M \in \mathscr{R}(N)$ than N_1 and N_2 are obviously bisimilar (t is always enabled in both nets).

On the other hand, if there is $M \in \mathscr{R}(N)$, $M(p) = 0$, then Player 1 can reach M in N_1, forcing Player 2 to do the same in N_2 (the labelling is one-to-one). Then he can fire t in N_1 and wins since t is not enabled in N_2. Hence N_1, N_2 are not bisimilar in this case. □

The following remark mentions some related facts.

Remark 4.4. 1. The problem, whether a given net is deterministic, can be reduced to the coverability problem (and is at least as hard) - we do not need "exact" reachability but only the reachability of a covering (componentwise greater or equal) marking.

2. The problem, whether a given net is deterministic up to bisimilarity, can be reduced to the reachability problem: we can take two copies of the given net and construct the above described game net. Player 1 can win iff the original net was not deterministic up to bisimilarity. Using *SPZRP* (cf. the proof of the previous lemma), it can be shown that this problem is at least as hard as the reachability problem.

3. Labelled Petri nets can be considered as a special case of finitely branching transition systems. For them, non-bisimilarity is semi-decidable (cf. e.g. [20] and [5]); hence semi-decidability of bisimilarity is sufficient to establish decidability. Therefore we can consider another, somewhat artificial, generalization. If, in the case of bisimilar nets, there is a (defending) strategy for Player 2 controlled by a finite automaton (it inputs the transition fired by Player 1 and outputs the transition for Player 2 to fire) then bisimilarity is semi-decidable: we can generate all finite automata successsively, incorporating each of them in the game net and checking whether Player 1 has a possibility to win (if not, the nets are bisimilar).

As promised, we finish this subsection by recalling a decidable generalization of the reachability problem which also comprises reachability of winning situations in game nets. The following auxiliary definition and lemma are taken from [15]:

Definition 4.5. (*Jančar* [15, *Definition* 5.1]). Let $\Sigma = (P,T,F)$ be a net. *Language* L_Σ is the set of formulas defined as follows:

(1) there is one variable $\mathscr{M}$ for elements of $\mathscr{N}^P$;

(2) a *term* is either atomic – $\mathscr{M}(p)$ or c, where $p \in P, c \in \mathscr{N}$ – or of the form $t_1 + t_2$, where t_1, t_2 are terms;

(3) a *formula* is either atomic – $t_1 < t_2$ or $t_1 \leqslant t_2$, where t_1, t_2 are terms – or is of the form $f_1 \& f_2$, where f_1, f_2 are formulas. The semantics is natural.

For a concrete marking M, $f(M)$ denotes the instance of f in which M is substituted for $\mathscr{M}$.

Lemma 4.6 (Jančar [15, Lemma 5.2]). *There is an algorithm with the following specification:*
Input: *A Petri net* $N = (\Sigma, M_0)$ *and a formula* $f \in L_\Sigma$,
Output: *YES if there is* $M \in \mathscr{R}(N)$ *s.t.* $f(M)$ *is true, NO otherwise.*

It can be easily verified that we can extend L_Σ by formulas $\neg f$ and $f_1 \vee f_2$ without losing the decidability. Note that formulas like $M \xrightarrow{t}, \neg\, M \xrightarrow{t}$ are easily expressible in (the extended) L_Σ.

It is easy to see that the conditions on Player 1's winning marking can be expressed in the extended L_Σ; therefore the reachability of such a winning marking is decidable.

4.2. Semilinear bisimulations

For our aims, we can suppose nets where each transition t has at least one input place p ($F(p,t) \geqslant 1$); if not, we can always add such p with 1 token and arcs (p,t), (t,p). Then adding a net N' with the zero initial marking to a net N (i.e. putting N and N' beside each other) has no effect – the resulting net is bisimilar to N.

Therefore we can restrict our considerations of bisimilarity to the pairs of Petri nets with the same underlying nets (they differ in initial markings only).

Then a bisimulation is, in fact, a relation on $\mathscr{N}^n$ for the relevant n. An equivalence relation R on $\mathscr{N}^n$ will be called a *congruence* if $(u,v) \in R$ implies $(u+w, v+w) \in R$ for any $w \in \mathscr{N}^n$ (addition taken componentwise).

As mentioned in the Introduction, the recent result of Christensen et al. [4] shows, in fact, that bisimilarity is decidable for the class of Petri nets where the bisimulation equivalence (the greatest bisimulation) is a congruence.

We extend this result using the notion of semilinear sets (cf. e.g. [8]).

Definition 4.7. A *set* $B \subseteq \mathscr{N}^k$ of k-dimensional nonnegative vectors is *linear* if there are vectors b (*basis*), $c_1, c_2, \cdots, c_n$ (*periods*) from $\mathscr{N}^k$ such that $B = \{b + x_1c_1 + x_2c_2 + \cdots + x_nc_n \mid x_i \in \mathscr{N}, 1 \leqslant i \leqslant n\}$. B is a *semilinear set* if it is a finite union of linear sets.

We will say that a relation on $\mathscr{N}^n$ is semilinear if it is semilinear as a subset of $\mathscr{N}^{2n}$.

Theorem 4.8. *For the class of (pairs of) labelled Petri nets where bisimilarity implies the existence of a semilinear bisimulation relating the initial markings, bisimilarity is decidable.*

Proof. As mentioned in the Remark in 4.4 (3), semidecidability is sufficient for establishing the decidability.

Note that there surely is an effective enumeration $B_0, B_1, B_2 \dots$ of all semilinear sets. Hence the scheme of the appropriate (partial) algorithm can be as follows:

Given $N_1 = (\Sigma, M_1)$, $N_2 = (\Sigma, M_2)$, perform the next cycle
(which can be infinite)
for $i = 0, 1, 2, \dots$ **do**
  if
(*) B_i is a bisimulation w.r.t. Σ relating the markings M_1, M_2
  then
    HALT (N_1, N_2 are bisimilar)
  endif

The crucial point is to show that the condition (*) can be verified effectively. For a fixed B_i, we first check that $B_i \subseteq \mathcal{N}^r \times \mathcal{N}^r$ where r is the number of places in Σ. Then (*) can be rewritten as follows:

$$(M_1, M_2) \in B_i$$
$$\bigwedge \ \forall (x,y) \in B_i \ [\forall a \forall x' \ (\ (x \xrightarrow{a}_{\Sigma} x') \Longrightarrow \exists y' (y \xrightarrow{a}_{\Sigma} y' \wedge (x', y') \in B_i))]$$
$$\bigwedge \ \forall (x,y) \in B_i \ [\forall a \forall y' \ (\ (y \xrightarrow{a}_{\Sigma} y') \Longrightarrow \exists x' (x \xrightarrow{a}_{\Sigma} x' \wedge (x', y') \in B_i))]$$

where $\forall a$ means for all a occurring as transition labels in Σ.

Since B_i is semilinear, there is a straightforward transformation of the formula into the Presburger arithmetic (theory of addition), which is known to be decidable (cf. e.g. [22]). □

Remark 4.9. The proof scheme offers an obvious generalization. In Theorem 4.8 the expression "a semilinear bisimulation" could be replaced by "a bisimulation from $\mathcal{B}$" where $\mathcal{B}$ is an effectively generable class of relations and where it is (semi)decidable for a given relation from $\mathcal{B}$ whether it is a bisimulation relating the initial markings.

Besides, it could be formulated in terms of (general) transition systems instead of Petri nets. Nevertheless it is not important for our aims here.

The fact that Theorem 4.8. is an extension of the mentioned result of [4] follows from the next theorem proved in [7]. (In our case, the relevant monoid is the set $\mathcal{N}^n$ with (vector) addition.)

Theorem 4.10 (Eilenberg and Schützenberger [7], Theorem II). *Every congruence in a finitely generated commutative monoid M is a rational (or semilinear) subset of $M \times M$.*

It is easy to construct a net for which the bisimulation equivalence is semilinear and is not a congruence. E.g., we can take $\Sigma = (\{p\}, \{t\}, F, L)$ where $F(p,t) = 2$ and $F(t,p) = 0$; the respective bisimulation equivalence is the set $\{(x,x) \mid x \in \mathcal{N}\} \cup \{(x, x+1) \mid x = 2k, k \in \mathcal{N}\} \cup \{(x+1, x) \mid x = 2k, k \in \mathcal{N}\}$.

5. Additional remarks

Petri nets are closely related to *vector addition systems*, VASs (introduced in [18]); n-dimensional VASs are isomorphic to (the reachability sets of) Petri nets with n places without self-loops (without both (p,t), (t,p) as arcs).

Hopcroft and Pansiot in [12] introduce *vector addition systems with states*, VASSs; any n-dim VASS can be viewed as a Petri net with (at most) n unbounded places. They show that any 2-dim VASS (unlike 3-dim) and any 5-dim VAS (unlike 6-dim) is an effectively computable semilinear set; hence the equality problem is decidable for them. (The complexity of the problem for 2-dim VASSs is studied in [13].) Ref. [12] also shows how any n-dim VASS can be simulated by an $(n+3)$-dim VAS; it can be done by a Petri net with $n+2$ places (using self-loops).

The proof of Theorem 3.7 can be easily modified (cf. Remark 3.8) to show the undecidability for (restricted subclasses of) 5-dim VASSs and 8-dim VASs. Thus the problems for dimensions 3,4 (VASSs) and 6,7 (VASs) remain open.

For bisimilarity and language equivalence we have undecidability for (a restricted subclass of) Petri nets with 2 unbounded places. For the case with 1 unbounded place, I conjecture that the bisimulation equivalence is semilinear and that both problems are decidable. In this sense, the decidability "border" would be established precisely.

But there are other kinds of the border. As mentioned, Hirshfeld in [11] shows that BPP-like Petri nets are sufficient for the undecidability of language equivalence; on the contrary, bisimilarity is decidable for them ([4]). It is also not difficult to modify the proofs in this paper in order to show the undecidability results for self-loop free Petri nets, and for single arcs only ($F(x,y) \leqslant 1$). Better exploring the relation of deterministic nets and "semilinear bisimulations" might help to understand the decidability border, too.

We considered the "classical" bisimilarity in the paper. In the literature, also other notions of bisimilarity, based on non-iterleaving semantics, have been defined and studied – step bisimulation, partial-word bisimulation, pomset bisimulation etc. (cf. e.g. [2]). The construction from the proof of Theorem 3.3 can be used to show their undecidability, too. The situation with the decidability results is probably more complicated.

We have mentioned the correspondence of Subsection 4.1 to some results in [29]. Vogler's notion *$\mathscr{F}$-deterministic net* [29, Definition 4.2.1] corresponds to the Petri net which is *deterministic up to weak bisimilarity* (cf. Remark 4.2). He shows that if one of two nets has the property then it is decidable whether they are failure equivalent [29, Theorem 4.2.9], which is in this case the same as being (weakly) bisimilar (this can be derived from Theorem 4.2.4 in [29]). Decidability of $\mathscr{F}$-determinism is shown as well (Theorem 4.2.7 in [29]).

Having an undecidable problem, we could explore its degree of undecidability. Due to finitely branching underlying transition systems, when considering (strong) bisimilarity for labelled Petri nets, we easily get Π_1^0-completeness of this problem (the problem whether two Petri nets are non-bisimilar is Σ_1^0-complete). [17] shows that the problem for *weak* bisimilarity is much more complicated from this point of view – it is beyond the arithmetical hierarchy. The upper bound Σ_1^1 (the first level of the analytical hierarchy) is clear from the definition (cf. e.g. [6]).

Acknowledgement

I would like to thank J. Bradfield, P. Čížek, Y. Hirshfeld and C. Stirling for helpful discussions; special thanks to S. Christensen whose observation helped to simplify the construction used in the undecidability proof.

I also thank the referees for helpful remarks and for drawing my attention to $\mathscr{F}$-deterministic nets of [29] and to paper [6].

References

[1] C. Autant, Z. Belmesk and Ph. Schnoebelen, Strong bisimilarity on nets revisited, in: *Proc. PARLE'91*, Lecture Notes in Computer Science, Vol. 506 (Springer, Berlin, 1991) 295–312.

[2] C. Autant and Ph. Schnoebelen, Place bisimulations in Petri nets, in: *Proc. 13th Int. Conf. Applications and Theory of Petri Nets*, Lecture Notes in Computer Science, Vol. 616 (Springer, Berlin, 1992) 45–61.

[3] H. Baker, Rabin's proof of the undecidability of the reachability set inclusion problem of vector addition systems, MAC – Memo 79, MIT, 1973.

[4] S. Christensen, Y. Hirshfeld and F. Moller, Bisimulation equivalence is decidable for all basic parallel processes, in: *Proc. CONCUR'93*, Lecture Notes in Computer Science, Vol. 715 (Springer, Berlin, 1993) 143–157.

[5] S. Christensen, H. Hüttel and C. Stirling, Bisimulation equivalence is decidable for all context-free processes, in: *Proc. CONCUR'92*, Lecture Notes in Computer Science, Vol. 630 (Springer, Berlin, 1992) 138–147.

[6] Ph. Darondeau, Bisimulation and effectiveness, *Inform. Process. Let.* **30** (1989) 19–20.

[7] S. Eilenberg and M. Schützenberger, Rational sets in commutative monoids, *J. Algebra* **13** (1969) 173–191.

[8] S. Ginsburg and E. Spanier, Semigroups, Presburger formulas, and languages, *Pacific J. Math.* **16** (1966) 285–296.

[9] M. Hack, Decision problems for Petri nets and vector addition systems, MAC Tech. Memo 53, MIT, 1975.

[10] M. Hack, The equality problem for vector addition systems is undecidable, *Theoret. Comput. Sci.* **2** (1976) 77–95.

[11] Y. Hirshfeld, Petri nets and the equivalence problem, in: *Proc. Computer Science Logic '93*, Lecture Notes in Computer Science, Vol. 832 (Springer, Berlin, 1994) 165–174.

[12] J. Hopcroft and J. Pansiot, On the reachability problem for 5-dimensional vector addition systems, *Theoret. Comput. Sci.* **8** (1979) 135–159.

[13] R. Howell, L. Rosier, D. Huynh and H. Yen, Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states, *Theoret. Comput. Sci.* **46** (1986) 107–140.

[14] H. Hüttel, Undecidable equivalences for Basic Parallel processes, in: *Proc. 13th Conf. on Foundations of Software Technology and Theoret. Comput. Sci.*, Bombay, India, 1993. (Also Tech. Report ECS-LFCS-93-276, Univ. of Edinburgh, UK, August 1993.)

[15] P. Jančar, Decidability of a temporal logic problem for Petri nets, *Theoret. Comput. Sci.* **74** (1990) 71–93.

[16] P. Jančar, Decidability questions for bisimilarity of Petri nets and some related problems, Tech. Report ECS-LFCS-93-261, Univ. of Edinburgh, UK, April 1993.

[17] P. Jančar, High undecidability of weak bisimilarity for Petri nets, in: *Proc. TAPSOFT/CAAP '95*, Lecture Notes in Computer Science, Vol. 915 (Springer, Berlin, 1995) 349–363.

[18] R. Karp and R. Miller, Parallel program schemata, *J. Comput. System Sci.* **3** (1969) 167–195.

[19] E. Mayr, An algorithm for the general Petri net reachability problem, *SIAM J. Comput.* **13** (1984) 441–460.

[20] R. Milner, *Communication and Concurrency* (Prentice Hall, Englewood Cliffs, NJ, 1989).

[21] M. Minsky, *Computation: Finite and Infinite Machines* (Prentice Hall, Englewood Cliffs, NJ, 1967).

[22] D.C. Oppen, A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger Arithmetic, *J. Comput. System Sci.* **16** (1978) 323–332.

[23] J.L. Peterson, *Petri Net Theory and the Modeling of Systems* (Prentice Hall, Englewood Cliffs, NJ, 1981).

[24] W. Reisig, *Petri Nets: an Introduction*, EATCS Monographs on TCS, Vol. 4 (Springer, Berlin, 1985).

[25] J. Shepardson and H. Sturgis, Computability of recursive functions, *J. Assoc. Comput. mach.* **10** (1963) 217–255.

[26] C. Stirling, Modal and temporal logics for processes, Lecture Notes to the Summerschool in Logical Methods in Concurrency, Aarhus University, Denmark, 1993.

[27] R. Valk and G. Vidal-Naquet, Petri nets and regular languages, *J. Comput. System Sci.* **23** (1981) 299–325.

[28] R.J. van Glabbeek, The linear time – branching time spectrum, in: *Proc. CONCUR'90*, Lecture Notes in Computer Science, Vol. 458 (Springer, Berlin, 1990) 278–297.

[29] W. Vogler, *Modular Construction and Partial Order Semantics of Petri Nets*, Lecture Notes in Computer Science, Vol. 625 (Springer, Berlin, 1992).