High undecidability of weak bisimilarity for Petri nets

Petr Jančar*

Dept. of Computer Science, University of Ostrava

Dvořákova 7, 701 00 Ostrava 1, Czech Republic

e-mail: jancar@oudec.osu.cz

August 8, 1994

Abstract

It is shown that the problem whether two labelled place/transition Petri nets (with initial markings) are weakly bisimilar is highly undecidable – it resides at least at level ω of the hyperarithmetical hierarchy. On the other hand it is in Σ_1^1 (the first level of the analytical hierarchy); the question whether the problem is in Δ_1^1 (i.e. whether it is hyperarithmetical) is left open here.

1 Introduction

In the theory of parallelism and concurrency, much effort is devoted to the decidability questions for various classes of (models of) processes and various notions of behavioural equivalences. It is often one of the first questions to be asked when developing automatic verification techniques.

This paper is partly motivated by recent results for the process algebra BPP (Basic Parallel Processes, cf. e.g. [CHM93]). The abstract syntax of a BPP-expression is

$$E ::= 0 \mid a.E \mid E + E \mid E \| E$$

The operational semantics is standard – 0 is a nil-process, $a.E$ performs the action a and then behaves like E, the operator $+$ means nondeterministic choice and $\|$ parallel merging (no communication between processes is allowed). A BPP-process is a family of equations $X_1 = E_1, \ldots, X_n = E_n$ where X_1 is the leading

*Supported by the Grant Agency of Czech republic, Grant No. 201/93/2123

variable (the initial state of the system). [CHM93] show that one of the central behavioural equivalences – the bisimulation equivalence (or bisimilarity, cf. [M89]) is decidable for BPP (in fact, the decidability was first established for a subclass – the normed processes and later extended for all BPP). Recently, [HJM94b] showed a polynomial algorithm for the normed BPP (the complexity for all BPP remaining open).

Remark. Although using different techniques, the results for BPP resemble very much those for BPA (Basic Process Algebra; cf. [CHS92], [HJM94a]).

BPP, in the considered sense (w.r.t. the corresponding transition system) can be viewed as a subclass of Petri nets – the well-known device for modelling parallel processes; every transition has exactly one input place in this case. Unfortunately, for the whole class of Petri nets, bisimilarity becomes undecidable ([J94]). The construction in the undecidability proof was cleverly modified in [Hi93] to show the undecidability of language equivalence for BPP. [Hü93] then added another modification to show the undecidability for other equivalences in the linear time – branching time spectrum of [G90]. On the other hand, [KH94] recently showed the decidability procedures for some finer versions of bisimilarity.

Allowing nonobservable, internal, actions (mostly denoted by τ), weak versions of the mentioned equivalences can be defined. For practice, these are more important, unfortunately the decidability is often affected; from finitely branching systems, we can get infinitely branching ones and it surely makes things less feasible.

In finitely branching systems, the problem of non-bisimilarity (non-equivalence) is usually easier than that of bisimilarity (equivalence); we can simply construct a tree of all possibilities, which will reveal eventually if the systems are non-bisimilar – hence we have semi-decidability of the negative case. In the positive case, a similar finite witness of equivalence often does not exist. If it does exist, it is then much more demanding to show it; [CHM93] succeeded in proving it for BPP and bisimilarity (similarly [CHS92] for BPA).

In case of weak bisimilarity, the situation seems to be different. The decidability question for BPP is still open ([KH94] show decidability for weak versions of the above mentioned finer equivalences on a subclass of BPP). But [E94] shows that a finite witness of the *positive* case exists – in this way establishing semidecidability of weak bisimilarity for BPP. It is the *negative* case which is not clear now; the task to show the existence of a finite witness for (weakly) non-bisimilar BPP's has not been accomplished. One thing is clear: if it should be undecidable (i.e. the negative case being non-semidecidable), it can not be shown by reducing the halting problem to the negative case, which was, in principle, the method of showing the undecidability results mentioned above.

We can ask the question in the frame of the whole class of Petri nets. Of course, undecidability of (strong) bisimilarity implies undecidability for weak bisimilarity but it still could be the case that weak non-bisimilarity is semidecidable. (In the strong case the nonbisimilarity problem is Σ_1^0-complete). It is not unimaginable at first sight that nonbisimilarity (the winning strategy of Player 1 in the bisimulation game mentioned in Section 3) could be demonstrated in some reasonable sense; the positive answer would settle the problem with BPP as well. Unfortunately, we show here that it is not the case. Weak non-bisimilarity on Petri nets is not only non-semidecidable but is beyond the arithmetical hierarchy (!). Maybe this explains a bit why the semidecidability of weak non-bisimilarity for BPP (which hopefully is the case) is not so easy to prove.

Although negative, the result might have value on its own. From theoretical point of view, we might not be satisfied with the fact that a problem is undecidable but want to know "how much" undecidable it is. Such results can also tell us something about the notions of equivalences, can compare them in some sense. E.g. the language nonequivalence for Petri nets is Σ_1^0-complete for the strong as well as weak version (it follows from decidability of the reachability problem for Petri nets); the same is true for the language nonequivalence on BPP. The strong nonbisimilarity is Σ_1^0-complete for Petri nets and decidable for BPP. And now weak bisimilarity which is "almost" is decidable for BPP (hopefully without "almost") is not arithmetical for Petri nets; does it illustrate the power of the (internal) communication ?

Given a lower bound, it is natural to ask for an upper bound. The problem of weak bisimilarity for Petri nets is easily shown to be in Σ_1^1 (the first level of the analytical hierarchy); hence the negative formulation is in Π_1^1. Whether the problems belong to the intersection Δ_1^1 (i.e. whether they are hyperarithmetical) is left open.

The paper is organized as follows. Section 2 gives basic definitions and results. The proofs are given in Section 3. A construction from [J94], which is needed here, is described in Appendix (to make this paper self-contained).

2 Definition and results

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

A labelled net is a tuple $\Sigma = (P, T, F, L)$ where P and T are finite disjoint sets of *places* and *transitions* respectively, $F : (P \times T) \cup (T \times P) \longrightarrow \{0, 1\}$ is a *flow function* (for $F(x, y) = 1$, there is an *arc* from x to y; we do not use multiple arcs in the paper) and $L : T \longrightarrow A$ is a *labelling* (attaches an action name – from a set A – to each transition). We can suppose a fixed (countable) A containing a special, *unobservable*, action τ, the other actions being *observable*.

By *a Petri net* we mean a tuple $N = (\Sigma, M_0)$, where Σ is a labelled net and M_0 the *initial marking*; a *marking* M being a function $M : P \longrightarrow \mathcal{N}$. (A

marking gives the number of *tokens* for each place). A transition t is *enabled* at a marking M, $M \xrightarrow{t}_\Sigma$, if $M(p) \geq F(p,t)$ for every $p \in P$. An enabled transition t may *fire* at a marking M yielding marking M', $M \xrightarrow{t}_\Sigma M'$, where $M'(p) = M(p) - F(p,t) + F(t,p)$ for all $p \in P$.
In the natural way, the definitions can be extended for sequences of transitions $\sigma \in T^*$.

For any $a \in A$ (including τ), by $M \xrightarrow{a}_\Sigma M'$ we mean that $M \xrightarrow{t}_\Sigma M'$ for some t with $L(t) = a$; by $M \stackrel{a}{\Longrightarrow}_\Sigma M'$ we mean that $M \xrightarrow{\sigma}_\Sigma M'$ 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).

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 *weak bisimulation* if for all $(M_1, M_2) \in R$:
– for every a and M_1' s.t. $M_1 \stackrel{a}{\Longrightarrow}_{\Sigma_1} M_1'$, there is M_2' s.t. $M_2 \stackrel{a}{\Longrightarrow}_{\Sigma_2} M_2'$, where $(M_1', M_2') \in R$
and conversely
– for every a and M_2' s.t. $M_2 \stackrel{a}{\Longrightarrow}_{\Sigma_2} M_2'$, there is M_1' s.t. $M_1 \stackrel{a}{\Longrightarrow}_{\Sigma_1} M_1'$, where $(M_1', M_2') \in R$.

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

(If the relation $\stackrel{a}{\Longrightarrow}$ is replaced by $\xrightarrow{a}$, we get the notion of strong bisimilarity.)

Let WB denote the set of (Gödel numbers, codes od notations of) all pairs of Petri nets (N_1, N_2) which are weakly bisimilar. Similarly we use NWB for the set of all weakly non-bisimilar pairs.

We show the following upper-bound and lower-bound theorems. The details of the (hyper)arithmetical and analytical hierarchies can be found e.g. in [B89].

Theorem 1.

WB is in Σ^1_1 (the 1st level of the analytical hierarchy); hence NWB is in Π^1_1.

Theorem 2.

WB and NWB reside at least at the level ω in the hyperarithmetical hierarchy.

Theorem 1 is easy to establish regarding the definition of weak bisimilarity. Proof of Theorem 2 shows an algorithm which, given a 1st order formula ϕ of the standard structure $(\mathcal{N}, 0, S, +, \cdot)$, constructs two Petri nets which are weakly bisimilar iff ϕ is true. This gives the result since the set of true 1st order sentences of arithmetics lies at level ω in the hyperarithmetical hierarchy (cf. e.g. [B89, p. 560]).

3 Proofs

We start by the short proof of the upper bound.

Proof of Theorem 1.

By definition,

$$
\begin{array}{c}
\text{(the code of) a pair } (\Sigma_1, M_1), (\Sigma_2, M_2) \text{ is in } WB \\
\text{iff} \\
\text{there is a set } X \subseteq \mathcal{N}^{r_1} \times \mathcal{N}^{r_2} \text{ (where } r_i \text{ is the number of place in } \Sigma_i \text{) such that} \\
(M_1, M_2) \in X \quad \bigwedge \\
\bigwedge \quad \forall (x,y) \in X \; [\; \forall a \forall x' \; (\; (x \stackrel{a}{\Longrightarrow}_{\Sigma_1} x') \longrightarrow \exists y' (y \stackrel{a}{\Longrightarrow}_{\Sigma_2} y' \wedge (x', y') \in X) \;) \;] \quad \bigwedge \\
\bigwedge \quad \forall (x,y) \in X \; [\; \forall a \forall y' \; (\; (y \stackrel{a}{\Longrightarrow}_{\Sigma_2} y') \longrightarrow \exists x' (x \stackrel{a}{\Longrightarrow}_{\Sigma_1} x' \wedge (x', y') \in X) \;) \;]
\end{array}
$$

where $\forall a$ means for all a which occur in Σ_1 and/or Σ_2.

Because the relation $\stackrel{a}{\Longrightarrow}$ surely is arithmetical (it is even recursive due to decidability of the reachability problem), it can be easily verified that there is a 2nd order formula $\phi(x, X)$ with one free individuum variable x, one free set variable X and with no set quantifiers such that for all (codes) $c \in \mathcal{N}$:

$$c \in WB \Longleftrightarrow \exists X \phi(c, X)$$

Thus the desired upper bound is established. $\square$

Now we proceed with proving Theorem 2.

The main idea lies in showing a certain (algorithmic) construction $\mathcal{C}$ which can be used for any Minsky counter machine C ([M67]) with an even (for convenience) number of (nonnegative) counters $c_1, c_2, \ldots, c_{2n}$; the construction yields a pair of Petri nets N_1, N_2 such that

$$
\begin{array}{c}
N_1,\ N_2 \text{ are weakly bisimilar} \\
\text{iff} \\
\text{it is true that} \quad \exists x_1 \forall x_2 \ldots \exists x_{2n-1} \forall x_{2n} \neg ! C(x_1, x_2, \ldots, x_{2n})
\end{array}
$$

By the notation $\neg ! C(x_1, x_2, \ldots, x_{2n})$ we mean that C does not halt for input counter values $x_1, x_2, \ldots, x_{2n}$.

When we show the construction $\mathcal{C}$, the claim is obvious:

Proof of Theorem 2. (supposing the construction $\mathcal{C}$)

Given any 1st order formula ϕ, using symbols $0, S, +, \cdot$ besides the usual logical symbols, we can transform it into an equivalent formula in the form $\exists x_1 \forall x_2 \ldots \exists x_{2n-1} \forall x_{2n} \phi'(x_1, x_2, \ldots, x_{2n})$ for some n, ϕ' being quantifier-free. (It can be accomplished by a standard transformation into the prenex normal form and by using dummy variables.)

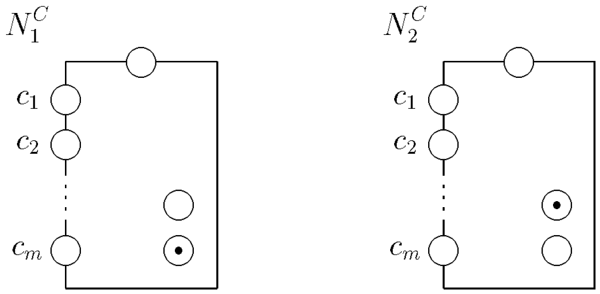

Figure 1:

Formula ϕ' is, in fact, a recursive predicate and it is a technical routine to construct a counter machine C with $2n$ counters such that for all $x_1, x_2, \ldots, x_{2n}$

$$\neg !C(x_1, x_2, \ldots, x_{2n}) \quad \text{iff} \quad \phi'(x_1, x_2, \ldots, x_{2n}) \text{ is true}$$

(in fact, C could need some additional counter(s); in this case we could use more dummy variables and then use the respective counters as the additional ones).

For C, we can construct N_1, N_2 by our (promised) construction $\mathcal{C}$ and we have

$$\begin{gathered}\phi \text{ is true in } (\mathcal{N}, 0, S, +, \cdot) \\ \text{iff} \\ \exists x_1 \forall x_2 \ldots \exists x_{2n-1} \forall x_{2n} \phi'(x_1, x_2, \ldots, x_{2n}) \text{ is true in } (\mathcal{N}, 0, S, +, \cdot) \\ \text{iff} \\ \exists x_1 \forall x_2 \ldots \exists x_{2n-1} \forall x_{2n} \neg !C(x_1, x_2, \ldots, x_{2n}) \text{ is true} \\ \text{iff} \\ N_1,\ N_2 \text{ are weakly bisimilar}\end{gathered}$$

by which the proof is finished. $\square$

Now we have to describe the construction $\mathcal{C}$.

As a basis, we use a result from [J94]. It shows a construction which, given a counter machine C with m counters, yields two nets N_1^C and N_2^C (cf. Fig. 1). The nets have the same static structure; places $c_1, c_2, \ldots, c_m$ for counters, a "start" place (the upper one in Fig.1), two special places (drawn inside the box) and some other places (and transitions, of course). They only differ in the (predefined) initial marking which is zero except for the two special places (as shown in Fig.1).

Remark. By labelling the transitions in N_1^C (and hence also in N_2^C) no τ-labels are used; therefore weak and strong bisimilarity coincide in this case.

The mentioned construction is fairly simple and is given in Appendix (to make this paper self-contained). What we need is the following proposition.

Proposition 3.

There is an algorithm which, given any counter machine C, constructs two Petri nets N_1^C, N_2^C, each containing a designated start place and places $c_1, c_2, \ldots, c_m$ where m is the number of counters of C, such that:
for any $x_1, \ldots, x_m$ and $x'_1, \ldots, x'_m$, if we put a token in the start place of both N_1^C and N_2^C, $x_1, \ldots, x_m$ tokens in places $c_1, \ldots, c_m$ of N_1^C and $x'_1, \ldots, x'_m$ tokens in $c_1, \ldots, c_m$ of N_2^C respectively then the resulting nets are (weakly) bisimilar iff $x_i = x'_i$ for all i, $1 \leq i \leq m$, and $\neg !C(x_1, \ldots, x_m)$.

Proof.

For the case with the same input counter values ($x_i = x'_i$ for all i), the construction from [J94] (or Appendix) satisfies.

When a counter machine C has different computations for any different inputs $x_1, \ldots, x_m$ and $x'_1, \ldots, x'_m$ then an easy inspection of the construction of [J94] shows that N_1^C and N_2^C are nonbisimilar when starting with different counter values (Player 1 in the below described bisimulation game forces the situation, when the counter to be tested is zero in one net and nonzero in the other; i.e. in one net a transition, with its unique label, is enabled while in the other net not). Therefore for such a machine C, the construction of [J94] is sufficient again.

To prove the proposition completely, it suffices to realize that for any C we can construct the following C' with an additional auxiliary counter: C' starts with "moving" the value of each counter, successively, into the auxiliary counter and back and then computes like C. Therefore C' really has different computations for any different inputs (and is equivalent to C w.r.t. halting). Hence the nets $N_1^{C'}$ and $N_2^{C'}$ in the "original" sense of [J94], with the place for the auxiliary counter viewed as "internal", can be taken as N_1^C, N_2^C in our proposition. $\square$

In the rest of this section, we suppose a fixed counter machine C with $2n$ counters $c_1, c_2, \ldots, c_{2n}$.

We first sketch the main idea informally. For such aims, it is useful to consider bisimilarity in terms of the following game.

(Weak) Bisimulation Game

1. *Prerequisities*
   There are two players, Player 1 and Player 2, and a pair of Petri nets N_1, N_2 (as the "playboard").

2. *Rules*
   Player 1 chooses one of the nets changes the marking (by firing a sequence of transitions) according to the relation $\stackrel{a}{\Longrightarrow}$ for some a.
   Then Player 2 replies by changing the marking in *the other* net according to $\stackrel{a}{\Longrightarrow}$ for the same a (if it is possible).
   Again, Player 1 chooses one of the nets ..., Player 2 replies ... 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 as successful for Player 2 (he defends successfully).

It is not difficult to see that Player 1 has a winning strategy if and only if N_1, N_2 are not weakly bisimilar.

Now consider another game.

Game G (for Player 1 and Player 2)

a/ *Board:* the nets N_1^C and N_2^C

b/ *Rules:*

1. At first, there are $2n$ special moves:
   – Player 2 begins and puts an arbitrary number of tokens x_1 in the places c_1 in both nets (the same value in both nets),
   – Player 1 continues and puts an arbitrary number of tokens x_2 in the places c_2.
   – Player 2 continues setting the value of c_3.
   ...
   – Player 1 sets the value of c_{2n}.

2. After these $2n$ moves, the usual bisimulation game starts (Player 1 begins, ...).

Proposition 3 ensures that Player 1 has no winning strategy in the game G (in other words: Player 2 has a defending strategy) iff it is true that $\exists x_1 \forall x_2 \dots \exists x_{2n-1} \forall x_{2n} \neg ! C(x_1, x_2, \dots, x_{2n})$.

Now we show that, based on N_1^C, N_2^C, we can build two nets in which the mentioned $2n$ moves are forced, in some sense (setting a value being acccomplished by τ-transitions, which is their "only" use).

Inductively, for $j = 2n, 2n-1, \dots, 1, 0$, we will construct nets N_j^L, N_j^R (L for "left", R for "right"). N_0^L, N_0^R will be the desired N_1, N_2. Intuitively, (starting) the pair N_j^L, N_j^R corresponds to the situation in the game G where j counters have been set.

We illustrate the construction in figures where we use broken arrows leading from a place to a (sub)net; the meaning is explained in Fig.2. Such an arrow leading from a place r to a (sub)net Σ and being labelled by a subset of places of Σ denotes the fact that there are special τ-labelled transitions; r is a "run-place" for these transitions and firing a sequence of them allows to add (or to subtract)

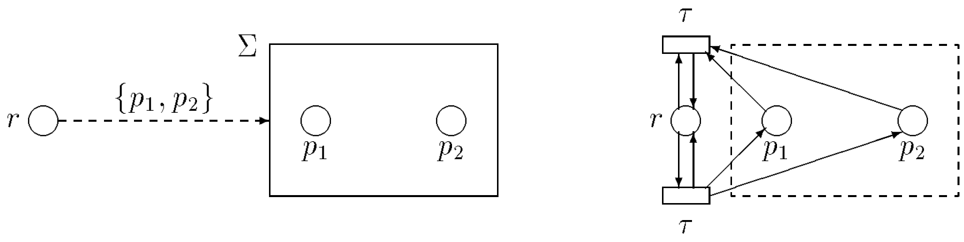

Figure 2:

an arbitrary – but the same for all – number to (or from) all places in the subset. Fig.2 illustrates it for a 2-element subset.

N_j^L and N_j^R will contain more "copies" of (places corresponding to) counters. By labelling a broken arrow by c_{2k+1}, for example, we understand that the label c_{2k+1} stands for the set of all copies of the counter c_{2k+1} in the target subnet.

The (inductive) construction of N_j^L and N_j^R

1. Let N_{2n}^L be N_1^C and let N_{2n}^R be N_2^C.
2. For an odd $j = 2k+1$, N_{2k+1}^L, N_{2k+1}^R are shown in Fig.3. (N_{2k+1}^L contains N_{2k+1}^R as a subnet).
3. For an even $j = 2k$, N_{2k}^L, N_{2k}^R are shown in Fig.4. (N_{2k}^R contains N_{2k}^L as a subnet).

The action a in the figures is observable (not τ). By the "start place" of N_j^L (or N_j^R) we mean the upper place in the figure.

Consider now the bisimulation game on the nets N_0^L and N_0^R with a token in the start place of both – and zero elsewhere with the exception of the special (internal) places of copies of N_1^C, N_2^C (recall Fig.1), which are subnets of the nets being considered.

In principle, the beginning of the game is as follows: Player 1 fires some τ-sequence and an a-transition in one net, Player 2 replies with firing a τ-sequence and an a-transition in the other net etc. – after $2n$ such moves of Player 1 (and $2n$ answers of Player 2), two from the copies of N_1^C, N_2^C are "started" (with some input values determined by previous τ-sequences). Notice that there is no sense for Player 1 nor Player 2 to fire in one move a τ-sequence only or to add some τ-sequence after firing an a – the results without and with such an τ-sequence are obviously weakly bisimilar (the effect of such a sequence can be achieved or "undone" freely by a τ-sequence starting the next move).

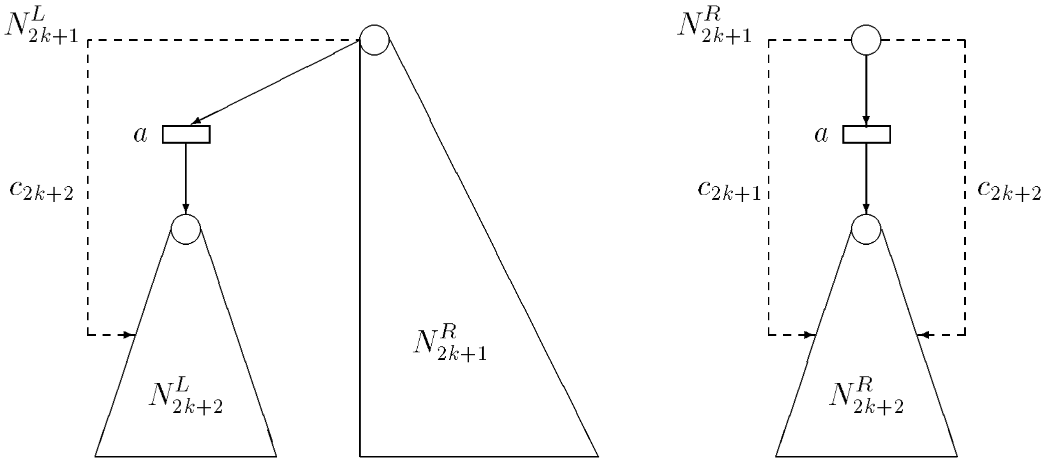

Figure 3: N_j^L and N_j^R for ODD $j = 2k + 1$

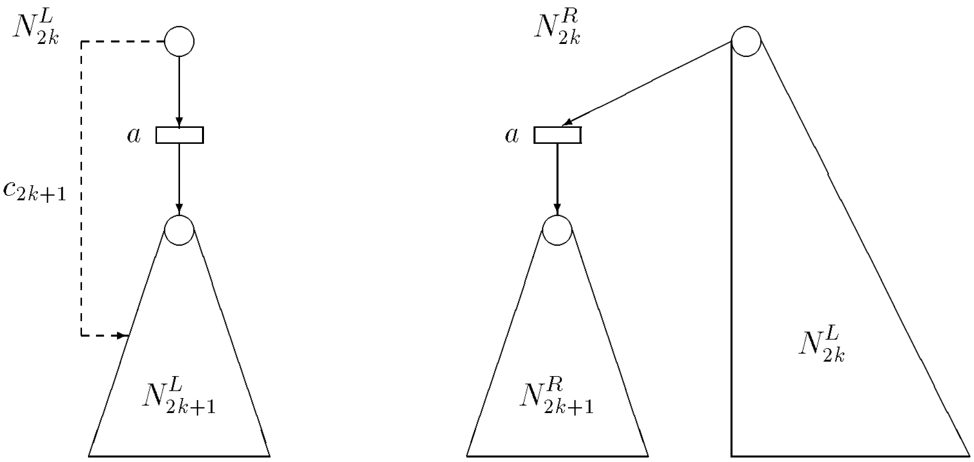

Figure 4: N_j^L and N_j^R for EVEN $j = 2k$

Now it suffices to prove the following propositions $\mathcal{P}_j$, $j = 2n, 2n-1, \ldots, 1, 0$. $\mathcal{P}_0$ shows that N_0^L, N_0^R are really our "desired" N_1, N_2 (relevant to our fixed C). We again deal with the cases "j even" and "j odd" separately.

Proposition $\mathcal{P}_{2k}$.

For any $x_1, x_2, \ldots, x_{2k}$ and $x'_1, x'_2, \ldots, x'_{2k}$, when we put a token in the start place of both N_{2k}^L and N_{2k}^R, $x_1, x_2, \ldots, x_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N_{2k}^L and $x'_1, x'_2, \ldots, x'_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N_{2k}^R respectively then the resulting nets are weakly bisimilar iff $x_i = x'_i$ for all $i = 1, 2, \ldots, 2k$ and $\exists x_{2k+1} \forall x_{2k+2} \ldots \exists x_{2n-1} \forall x_{2n} \neg ! C(x_1, x_2, \ldots, x_{2n})$.

Proposition $\mathcal{P}_{2k+1}$.

For any $x_1, x_2, \ldots, x_{2k}$, any $x'_1, x'_2, \ldots, x'_{2k}$ and any x_{2k+1}, when we put a token in the start place of both N_{2k+1}^L and N_{2k+1}^R, $x_1, x_2, \ldots, x_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N_{2k+1}^L, $x'_1, x'_2, \ldots, x'_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N_{2k+1}^R respectively and x_{2k+1} tokens in all copies of c_{2k+1} in N_{2k+1}^L then the resulting nets are weakly bisimilar iff $x_i = x'_i$ for all $i = 1, 2, \ldots, 2k$ and $\forall x_{2k+2} \exists x_{2k+3} \forall x_{2k+4} \ldots \exists x_{2n-1} \forall x_{2n} \neg ! C(x_1, x_2, \ldots, x_{2n})$.

First note that the base case, Proposition $\mathcal{P}_{2n}$, follows immediately from Proposition 3. (Recall that N_{2n}^L is N_1^C and N_{2n}^R is N_2^C.) Therefore to prove $\mathcal{P}_{2k}$ ($\mathcal{P}_{2k+1}$), we can suppose that $\mathcal{P}_{2k+1}$ ($\mathcal{P}_{2k+2}$) holds. We start with Proposition $\mathcal{P}_{2k}$ since it is a bit easier.

Proof of Proposition $\mathcal{P}_{2k}$

Consider N_{2k}^L and N_{2k}^R, $k < n$, (cf. Fig.4) with a token in the start place of both and $x_1, x_2, \ldots, x_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N_{2k}^L and $x'_1, x'_2, \ldots, x'_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N_{2k}^R respectively.

If $x_i \neq x'_i$ for some $i, 1 \leq i \leq 2k$, then Player 1 can "start" N_{2k+1}^R firing the "left" a-transition in N_{2k}^R after which Player 2 is forced to start N_{2k+1}^L – but not having the possibility to change the inequality $x_i \neq x'_i$, he reaches a "situation" which is non-bisimilar due to Proposition $\mathcal{P}_{2k+1}$.

Now suppose $x_i = x'_i$ for all $i, 1 \leq i \leq 2k$. If Player 1 wants to win, he has to choose the left a in N_{2k}^R – by choosing another a, Player 2 could obviously reach the identical situation in both nets (the same "active" subnets). Recall that it has no sense for Player 1 to add a τ-sequence and note that values of $c_1, c_2, \ldots, c_{2k}$ remained unchanged in (now active) N_{2k+1}^R. Player 2 answers by firing a in N_{2k}^L preceeded by setting an arbitrary value x_{2k+1} in (all copies of) c_{2k+1} in N_{2k+1}^L (which now becomes active). If there is some x_{2k+1} such that $\forall x_{2k+2} \exists x_{2k+3} \forall x_{2k+4} \ldots \exists x_{2n-1} \forall x_{2n} \neg ! C(x_1, x_2, \ldots, x_{2n})$ then Player 2 could choose such x_{2k+1} and reach a bisimilar situation; otherwise, no matter what he sets in c_{2k+1}, the resulting situation is non-bisimilar (both variants follow from Proposition $\mathcal{P}_{2k+1}$). □

Proof of Proposition $\mathcal{P}_{2k+1}$

Consider N^L_{2k+1} and N^R_{2k+1}, (cf. Fig.3) with a token in the start place of both, $x_1, x_2, \ldots, x_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N^L_{2k+1}, $x'_1, x'_2, \ldots, x'_{2k}$ tokens in all copies of $c_1, c_2, \ldots, c_{2k}$ in N^R_{2k+1} respectively and x_{2k+1} tokens in all copies of c_{2k+1} in N^L_{2k+1}.

The case with $x_i \neq x'_i$ for some $i, 1 \leq i \leq 2k$ is similar to that in the proof for $\mathcal{P}_{2k}$.

Now suppose $x_i = x'_i$ for all $i, 1 \leq i \leq 2k$. It is clear that if Player 1 wants to win, he has to choose the left a – now in N^L_{2k+1}, thus starting N^L_{2k+2}. But before this starting he can set any value x_{2k+2} in (all copies of) c_{2k+2} in N^L_{2k+2}. Player 2 answers in N^R_{2k+1} by setting the same values of (all copies of) c_{2k+1} and c_{2k+2} in N^R_{2k+2} as are set in N^L_{2k+2} (in fact, Player 2 sets the same value of c_{2k+1} as he chose in the "left" net in the previous move and sets the same value of c_{2k+2} as Player 1 has just chosen); note that if he had not set the same values, he reaches a nonbisimilar situation due to Proposition $\mathcal{P}_{2k+2}$. If now for all x_{2k+2} it holds that $\exists x_{2k+3} \forall x_{2k+4} \ldots \exists x_{2n-1} \forall x_{2n} \neg ! C(x_1, x_2, \ldots, x_{2n})$ we have surely arrived at a bisimilar situation; otherwise Player 1 could choose some x_{2k+2} which ensures arriving at a nonbisimilar situation (again due to induction hypothesis $\mathcal{P}_{2k+2}$). $\Box$

Thus we have finished the description and the correctness proof for the construction $\mathcal{C}$ and we are done.

Acknowledgements

I would like to thank J.Bradfield and J.Esparza for helpful discussions.

References

[B89] J. Barwise (ed.), *Handbook of Mathematical Logic* (North Holland, 5th printing 1989).

[CHM93] 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.

[CHS92] 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.

[E94] J. Esparza, On the uniform word problem for commutative context-free grammars, unpublished.

[G90] 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.

[Hi93] Y. Hirshfeld, Petri nets and the equivalence problem, to appear in: *Proc.*

Computer Science Logic '93 (1993).

[HJM94a] Y. Hirshfeld, M. Jerrum and F. Moller, A polynomial algorithm for deciding bisimilarity of normed context-free processes, Tech.rep. ECS-LFCS-94-286, Dept. of comp. sci., Univ. of Edinburgh, UK, March 1994.

[HJM94b] Y. Hirshfeld, M. Jerrum and F. Moller, A polynomial-time algorithm for deciding bisimulation equivalence of normed Basic Parallel processes, Tech.rep. ECS-LFCS-94-288, Dept. of comp. sci., Univ. of Edinburgh, UK, April 1994.

[Hü93] H. Hüttel, Undecidable equivalences for Basic Parallel processes, 13th Conf. on Foundations of Software Technology and Theoretical Computer Science, Bombay, India, 1993
(also Tech.rep. ECS-LFCS-93-276, Dept. of comp. sci., Univ. of Edinburgh, UK, August 1993).

[J94] P. Jančar P.: Decidability questions for bisimilarity of Petri nets and some related problems; in *Proc. STACS'94*, Lecture Notes in Computer Science, Vol. 775 (Springer, Berlin, 1994) 581–592
(also Tech.rep. ECS-LFCS-93-261, Dept. of comp. sci., Univ. of Edinburgh, UK, April 1993).

[KH94] A. Kiehn and M. Hennessy, On the decidability of non-interleaving process equivalences, in *Proc. CONCUR'94* (to appear).

[M89] R. Milner, *Communication and Concurrency* (Prentice Hall, 1989).

[M67] M. Minsky, *Computation: Finite and Infinite Machines* (Prentice Hall, 1967).

Appendix

Here we briefly show the construction from [J94] which, given a counter machine C, yields the nets N_1^C, N_2^C referred to throughout the paper.

A counter machine C with nonnegative counters $c_1, c_2, ..., c_m$ is a program

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

where $COMM_n$ is a $HALT$-command and $COMM_i$ $(i = 1, 2, ..., n-1)$ are commands of the following two types

1/ $c_j := c_j + 1;\ goto\ k$

2/ $if\ c_j = 0\ then\ goto\ k_1\ else\ (c_j := c_j - 1;\ goto\ k_2)$

assuming $1 \leq k, k_1, k_2 \leq n$, $1 \leq j \leq m$.

The set BS of *branching states* is defined as $BS = \{i \mid COMM_i$ is of the type $2\}$.

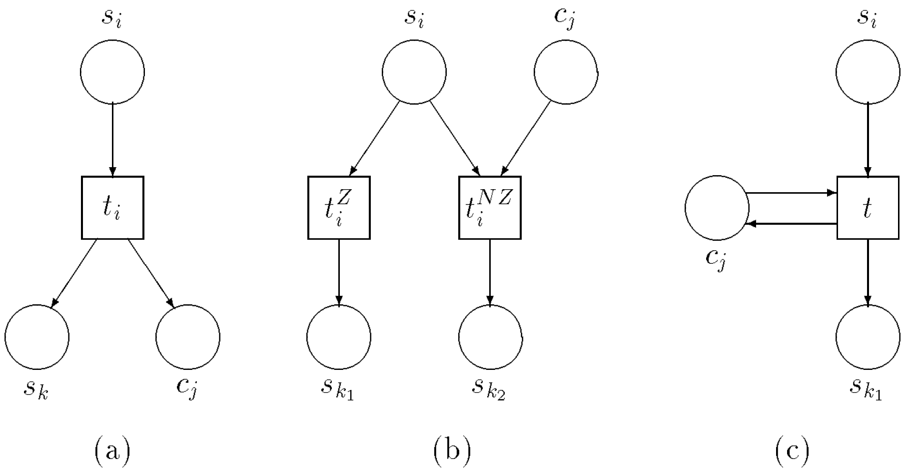

Figure 5:

Consider a fixed counter machine C in the above notation. First we describe a construction of the *basic net* N_C which simulates C in a weak sense.

Construction of N_C

1. Let $c_1, c_2, ..., c_m$ (the counter part) and $s_1, s_2, ..., s_n$ (the state part) be places of N_C .

2. For $i = 1, 2, ..., n-1$ 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. Figure 5(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 non-zero) with (s_i, t_i^{NZ}), (c_j, t_i^{NZ}), (t_i^{NZ}, s_{k_2}) (cf. Figure 5(b)).

3. The (predefined) initial marking is zero except for the place s_1 with 1 token.

N_C can simulate C in a natural way but (only) transitions t_i^Z can "*cheat*", i.e. fire although the relevant c_j is not 0.

Adding a *dc-transition* (*dc* for "definitely cheating") to N_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. Figure 5(c)).
Notice that such t has the same effect as t_i^Z but firing it always means cheating.

Based on N_C, we construct N_1^C, N_2^C as follows.

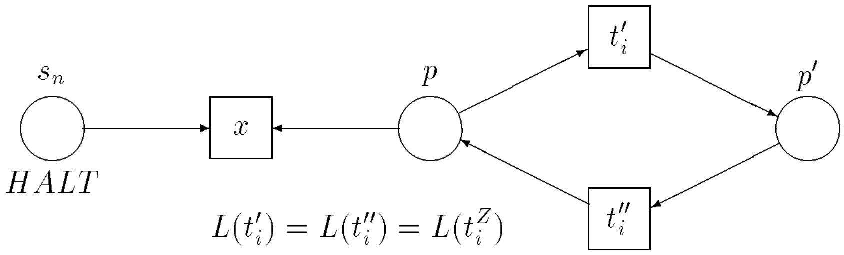

Figure 6:

Construction of N_1^C, N_2^C

1. To N_C, add new (special) places p, p' and a new transition x with arcs (s_n, x), (p, x).
2. 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) and additional arcs (p, t_i'), (t_i', p'), (p', t_i''), (t_i'', p) and put $L(t_i') = L(t_i'') = L(t_i^Z)$ (cf. Figure 6).
4. Now take two copies of the arised net.
   In one copy put 1 token in p and 0 in p', which yields N_1^C.
   In the other copy put 1 token in p' and 0 in p, which yields N_2^C.

Remark. It is shown in [J94] that for any $x_1, \ldots, x_m$:

$\neg !C(x_1, \ldots, x_m)$
iff
the nets arising from N_1^C, N_2^C by putting $x_1, \ldots, x_m$ in $c_1, \ldots, c_m$ are (strongly) bisimilar

If C halts for given $x_1, \ldots, x_m$ then Player 1 wins by firing the correct (noncheating) sequence in N_1^C finished by the special transition x. This does not work when C does not halt. And when Player 1 "cheats" then Player 2 can immediately reach the identical (hence bisimilar) situation in both nets.