Petri Nets and Regular Processes

Petr Jančar

Department of Computer Science, Technical University of Ostrava
17. listopadu 15, CZ-708 33 Ostrava-Poruba, Czech Republic

Javier Esparza

Institut für Informatik, Technische Universität München
Arcisstrasse 21, D-80290 München, Germany

Faron Moller

Computing Science Department, Uppsala University
P.O. Box 311, S-751 05 Uppsala, Sweden

Abstract

We consider the following problems: (a) Given a labelled Petri net and a finite automaton, are they equivalent? (b) Given a labelled Petri net, is it equivalent to some (unspecified) finite automaton? These questions are studied within the framework of trace and bisimulation equivalences, in both their strong and the weak versions. (In the weak version a special τ action – similar to a λ-transition in automata theory – is considered to be non-observable.) We demonstrate that (a) is decidable for strong and weak trace equivalence and for strong bisimulation equivalence, but undecidable for weak bisimulation equivalence. On the other hand, we show that (b) is decidable for strong bisimulation equivalence, and undecidable for strong and weak trace equivalence, and for weak bisimulation equivalence.

1 Introduction

In the specification and verification of distributed systems, it is typically the case that one considers a specific mathematical model for the description of processes, along with some equivalence relating processes which demonstrate the same semantic behaviour. One of the first questions to ask then for the purpose of (automatic) verification is that of (the extent of) the decidability of the equivalence.

In this paper we consider the class of processes generated by labelled place/transition Petri nets, called just Petri nets in the sequel. Petri nets constitute a popular and important formalism for modelling distributed systems, as exemplified by the widely-used textbooks by Peterson [20] and Olderog [19]. We consider trace equivalence and bisimulation equivalence – two equivalences in the forefront of the study of these systems – and study both their strong and weak versions.[1]

Unfortunately, already the strong versions (along with the strong versions of all 'reasonable' behavioural equivalences) are undecidable for general Petri nets [10, 11], in fact even for Petri nets having at most two unbounded places. Faced with such a negative result, a natural step then is to restrict the problem in some way. For example, for the class of Petri nets in which every transition has a single input place—the so-called Basic Parallel Processes— strong bisimulation equivalence is decidable [1], whereas all other standard equivalences (such as trace equivalence) are undecidable, even in the strong case [6, 8]. If on the other hand we compare two bounded Petri nets, then these equivalences all become decidable, as such nets describe behaviours depicted by finite automata.

We consider here the problem of restricting just one of the two Petri nets to be bounded, thus comparing general Petri nets against finite automata Within this framework, we consider both the equivalence problem, as well as the question concerning the finiteness of a given net, i.e., the question as to whether or not there is some (unspecified) finite automaton which is equivalent to the Petri net. We address these questions for both trace and bisimulation equivalence. [2] We show that the strong and weak trace equivalence problems are decidable, while the finiteness question for the traces of a net is undecidable, even in the strong case. In the bisimulation case, both the equivalence and regularity questions are decidable for strong bisimilarity, yet undecidable for weak bisimilarity.

Our results extend and complement previous results by Valk and Vidal-Naquet [21] on the finiteness question for trace equivalence, called the regularity question in [21]. They showed that the regularity of the terminal language of a net – i.e. the set of traces corresponding to the firing sequences leading to a fixed set of markings – is undecidable, whereas the regularity of the set of all traces of a net in which each transition carries a different label is decidable.

The paper is structured as follows. In Section 2 we define the concepts which we use, in particular the notion of a Petri net, as well as of the equivalences which we shall study. We also present several technical results—both old and new—which we shall exploit in our decision procedures and undecidability proofs. Of particular importance are results based on the decidability of the reachability problem for Petri nets, and relevant variations of Higman's Theorem.

In Section 3 we consider trace equivalence, and demonstrate first the decidability of the equivalence problem (in both the strong and weak cases) by demonstrating that the trace inclusion problem in each direction is decidable. We follow this by demonstrating the undecidability of the finiteness problem in the strong case. We show this by a reduction from the halting problem for Minsky machines.

[1]In the strong versions, all the labels carried by the transitions of the net are assumed to be visible actions. In the weak versions, some transitions may be labelled with a special silent action τ, which plays a similar rôle to λ-transitions in finite automata. The firing of these transitions is assumed to be unobservable.

[2]The finiteness problem has been called the regularity problem in the literature. This name is very adequate for trace equivalence for obvious reasons, but not so adequate for bisimulation equivalence.

In Section 4 we turn our attention to bisimulation equivalence, and demonstrate that both problems are decidable in the strong case, yet both problems are undecidable in general. The first undecidability result follows from a reduction from the containment problem for Petri nets, while the second relies on a special form of the containment problem to which the halting problem for Minsky machines can be reduced.

The results presented here elaborate on those presented by the authors in [13] and [15].

2 Preliminaries

Here we define some basic notions and introduce various results which will prove useful.

By $\mathbb{N}$ we denote the set of nonnegative integers $\mathbb{N} = \{0, 1, 2, \ldots\}$. For a set A, A^* denotes the set of finite sequences of elements of A; the empty sequence is denoted by $\lambda \in A^*$. For $u \in A^*$ and $k \in \mathbb{N}$, we denote by u^k the k-fold catenation of u.

2.1 Labelled Transition Systems and Equivalences

We shall define an automaton to be a *labelled transition system* (LTS), which is a tuple $\mathcal{L} = \langle S, \Sigma, \{\xrightarrow{a}\}_{a\in\Sigma} \rangle$ where S is a set of *states*, Σ is a finite set of *actions*, and each $\xrightarrow{a}$ is a binary *transition relation* on S, i.e. $\xrightarrow{a} \subseteq S \times S$; we shall write $E \xrightarrow{a} F$ for $\langle E, F \rangle \in \xrightarrow{a}$. By $E \longrightarrow F$ we mean that $E \xrightarrow{a} F$ for some a; and $\longrightarrow^*$ denotes the reflexive and transitive closure of the relation $\longrightarrow$. We shall write $E \xrightarrow{u} F$ for $u = a_1 a_2 \cdots a_n \in \Sigma^*$ to mean that there are states $E_1, E_2, \ldots, E_{n-1}$ such that $E \xrightarrow{a_1} E_1 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} E_{n-1} \xrightarrow{a_n} F$. We write $E \xrightarrow{u}$ to mean that $E \xrightarrow{u} F$ for some F. In particular, $E \xrightarrow{\lambda} E$ for every E, and $E \xrightarrow{\lambda} F$ only if $E = F$. (Note the difference between $E \longrightarrow F$ and $E \xrightarrow{\lambda} F$.)

We say that a set of states S' is *reachable* from E, written $E \longrightarrow^* S'$, iff $E \longrightarrow^* F$ for some $F \in S'$. The *reachability set* for a state E is defined by $\mathcal{R}(E) = \{F : E \longrightarrow^* F\}$.

An LTS $\mathcal{L} = \langle S, \Sigma, \{\xrightarrow{a}\}_{a\in\Sigma} \rangle$ is *finite-state* iff S is finite. $\mathcal{L}$ is *image finite* iff $succ_a(E) = \{ F : E \xrightarrow{a} F \}$ is finite for every $E \in S$ and every $a \in \Sigma$.

By a *process* E we refer to a state in a transition system; when necessary, we shall denote the underlying transition system by $\mathcal{L}(E)$. By referring to a *finite-state process* E, we mean that $\mathcal{L}(E)$ is finite; a similar convention holds for an *image finite process*. We shall use the symbols $R, R', \ldots$ to denote finite-state LTSs, and the symbols $r, r', \ldots$ to denote states in finite-state systems, that is, finite-state processes.

A binary relation $\mathcal{B}$ between processes is a *strong bisimulation* provided that whenever $\langle E, F \rangle \in \mathcal{B}$, for each $a \in \Sigma$,

- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $\langle E', F' \rangle \in \mathcal{B}$; and
- if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $\langle E', F' \rangle \in \mathcal{B}$.

Two processes E and F are *strongly bisimulation equivalent* or *strongly bisimilar*, written $E \sim F$, iff there is a strong bisimulation $\mathcal{B}$ relating them.

A decreasing chain $\sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots \supseteq \sim$ of equivalence relations between processes is defined inductively as follows.

1. $E \sim_0 F$ for all processes E and F;
2. $E \sim_{n+1} F$ iff for each $a \in \Sigma$,
   - if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $E' \sim_n F'$; and
   - if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $E' \sim_n F'$.

The fact that these relations do form a decreasing chain of equivalences all containing $\sim$ is easily confirmed. The next two Propositions are also easily-confirmed folklore.

Proposition 2.1 *For image finite processes, $E \sim F$ iff $E \sim_n F$ for all $n \geq 0$.*

Let us call $\mathcal{L} = \langle\, S, \Sigma, \{\xrightarrow{a}\}_{a\in\Sigma}\,\rangle$ an *admissible system* iff the state set S is finite or countably infinite (identified with a set of sequences over a finite alphabet), $\mathcal{L}$ is image finite, and all of the successor functions $succ_a : S \longrightarrow 2^S$ are effectively computable. (Recall that Σ is finite, so there are only finitely many of these.) With this restriction in place, the following result is immediate.

Proposition 2.2 *Considering only admissible systems, all of the relations $E \sim_n F$ are decidable. Therefore the nonequivalence problem $E \not\sim F$ is semidecidable.*

We now describe the background for a decision procedure which we shall employ later. Given a transition system $\mathcal{L} = \langle\, S, \Sigma, \{\xrightarrow{a}\}_{a\in\Sigma}\,\rangle$, we define the class of all n-*incompatible processes* (taken from other transition systems) as $\textsc{Inc}_n^{\mathcal{L}} = \{E \ : \ \forall F \in S : E \not\sim_n F\}$.

Proposition 2.3 *For any n, $E \sim F$ implies that $E \sim_n F$ and $E \not\longrightarrow^* \textsc{Inc}_n^{\mathcal{L}(F)}$. In addition, the reverse implication holds under the further proviso that $\sim_{n-1}$ coincides with $\sim_n$ (and hence with $\sim$) over $\mathcal{L}(F)$.*

Proof: The left-to-right implication is obvious. For the right-to-left implication, it is straightforward to verify that, assuming $\sim_{n-1} = \sim_n$ on $\mathcal{L}(F)$, the set

$$\Big\{\, \langle\, E', F'\,\rangle \ : \ E' \in \mathcal{L}(E),\ F' \in \mathcal{L}(F),\ E' \sim_n F',\ E' \not\longrightarrow^* \textsc{Inc}_n^{\mathcal{L}(F)} \,\Big\}$$

is a bisimulation. The crucial point to observe is that whenever we have that $E'' \sim_{n-1} F''$ and $E'' \notin \textsc{Inc}_n^{\mathcal{L}(F)}$ we must have that $E'' \sim_n F''$. $\Box$

Corollary 2.4 *For any two states r and r' of an n-state LTS R, $r \sim_{n-1} r'$ iff $r \sim_n r'$ (iff $r \sim r'$). Therefore, for any process E and any state r of R,*

$$E \sim r \quad \textit{iff} \quad E \sim_n r \ \textit{ and } \ E \not\longrightarrow^* \textsc{Inc}_n^R.$$

Proof: As $\sim_{i+1} \subseteq \sim_i$, and $\sim_i = \sim_{i+1}$ implies $\sim_i = \sim_{i+k}$ for any $k \geq 0$, these equivalence relations must stabilize within the first n-steps over any n-state LTS. □

Corollary 2.5 *To demonstrate the decidability of $E \sim r$ for any specified class of processes E for which $E \sim_n r$ is decidable, it suffices to demonstrate the decidability of the (non-)reachability problem $E \not\longrightarrow^* \textsc{Inc}_n^R$.*

Further development and applications of this technique are presented in [14].

We have as yet dealt only with definitions and results concerning automata without silent transitions. To introduce these transitions, we interpret a distinguished symbol $\tau \in \Sigma$ as a silent action, and modify our definitions accordingly. (We follow this framework adopted from process theory rather than the automata theoretic technique of directly allowing λ transitions as we want to be able to distinguish, for example, between $\stackrel{a}{\longrightarrow}$ and $\stackrel{\tau a}{\longrightarrow}$; whereas $\lambda a = a$, $\tau a \neq a$.)

For any action $a \in \Sigma$, by $E \stackrel{a}{\Longrightarrow} F$ we mean that $E \stackrel{u}{\longrightarrow} F$ for some $u = \tau^k a \tau^\ell$ $(k, \ell \geq 0)$; in the case where $a = \tau$, we also allow $u = \lambda$, recalling that $E \stackrel{\lambda}{\longrightarrow} F$ implies $E = F$, so for example $E \stackrel{\tau}{\Longrightarrow} E$ for all E. [3] The relations $E \stackrel{u}{\Longrightarrow} F$, $E \stackrel{u}{\Longrightarrow}$, where $u \in \Sigma^*$, are then the obvious generalizations.

The relation of *weak bisimulation equivalence*, denoted by $\approx$, as well as the relations $\approx_n$ $(n = 0, 1, 2, \ldots)$, are defined in the same way as for the strong relations $\sim$, $\sim_n$ but with $\stackrel{a}{\longrightarrow}$ replaced everywhere by $\stackrel{a}{\Longrightarrow}$.

The *strong trace set* of a state E of an LTS $\mathcal{L}$ is defined by $\mathcal{ST}(E) = \{w \in (\Sigma)^* : E \stackrel{w}{\longrightarrow}\}$. Two processes E and F are *strongly trace equivalent* iff $\mathcal{ST}(E) = \mathcal{ST}(F)$. The *weak trace set*, or just *trace set* of a state E is defined by $\mathcal{T}(E) = \{w \in (\Sigma \setminus \{\tau\})^* : E \stackrel{w}{\Longrightarrow}\}$. Two processes E and F are *weakly trace equivalent*, or just *trace equvalent*, iff $\mathcal{T}(E) = \mathcal{T}(F)$.

Notice that two τ-free transition systems are weakly trace equivalent iff they are strongly trace equivalent, and they are weakly bisimilar iff they are strongly bisimilar. As an easy consequence, decidability of a problem in the weak case implies decidability of the strong case. Moreover, undecidability of a problem in the strong case can be shown by proving undecidability in the weak case for τ-free systems. We shall make free use of these facts.

2.2 Petri Nets

A *(finite, labelled, place/transition Petri) net* is a tuple $N = \langle P, T, F, \Sigma, \ell \rangle$ where

- P, T and Σ are finite disjoint sets of *places*, *transitions* and *actions*, respectively;

[3] This is somewhat nonstandard in process theory; our relations $\stackrel{a}{\Longrightarrow}$ should be written as $\stackrel{\widehat{a}}{\Longrightarrow}$ in order to fit into the process theory framework [17], but in our presentation we omit the extra decoration.

- $F : (P \times T) \cup (T \times P) \longrightarrow \{0,1\}$ defines the set of *arcs*; $\langle\, x, y \,\rangle$ is an arc iff $F(x,y) = 1$;
- $\ell : T \longrightarrow \Sigma$ is a *labelling*, which associates an action from Σ to each transition.

In the Petri net literature, multiple arcs are often allowed (in which case the range of F is given as $I\!N$). For technical convenience, we treat only ordinary nets; nevertheless all of our arguments can be easily modified to hold for these more general nets.

A *marking* of a net is a mapping $M : P \longrightarrow I\!N$ associating a number of *tokens* to each place. A transition t is *enabled* at a marking M, written $M[t\rangle$, iff $M(p) \geq F(p,t)$ for every $p \in P$. If a transition t is enabled at a marking M it may *fire* or *occur* yielding the marking M', denoted $M[t\rangle M'$, where $M'(p) = M(p) - F(p,t) + F(t,p)$ for all $p \in P$.

We shall display nets graphically using circles for places and boxes for transitions. A marking for a place (the number of tokens) will be shown by a number (variable) or by black dots inside the circle. When labels of transitions are important, we write them inside the boxes.

We shall interpret a net $N = \langle\, P, T, F, \Sigma, \ell \,\rangle$ as an LTS where markings play the role of states. The transition relations $\xrightarrow{a}$ are provided by the firings of the enabled transitions of the net: $M \xrightarrow{a} M'$ iff $M[t\rangle M'$ for some t with $\ell(t) = a$. We also say $M \longrightarrow M'$ if $M \xrightarrow{a} M'$ for some a. Notions like $M \stackrel{u}{\Longrightarrow} M'$, $\mathcal{T}(M)$, $M_1 \sim M_2$ are then inherited from the respective notions given in the general setting.

We now recall some known results from Petri net theory, in particular the decidability of the reachability problem.

Theorem 2.6 [16] *Given two markings M and M' of a Petri net N, it is decidable whether or not $M \longrightarrow^* M'$, that is, whether or not $M' \in \mathcal{R}(M)$.*

We shall also use the notion of an ω-marking; it extends the notion of marking by allowing an infinite number of tokens to be associated to the places. Formally we set $I\!N_\omega = I\!N \cup \{\omega\}$ where we suppose ω satisfies $n \leq \omega$ and $\omega + n = \omega - n = \omega$ for all $n \in I\!N$. An *ω-marking*, for which we reserve symbols $\widehat{M}$, $\widehat{M}'$, ..., is then simply a mapping $\widehat{M} : P \longrightarrow I\!N_\omega$. Notions such as $\widehat{M} \xrightarrow{a} \widehat{M}'$ and $\mathcal{T}(\widehat{M})$ are then naturally defined as extensions of the previous definitions.

Taking the pointwise ordering $\leq$ on (ω-)markings, we observe the trivial monotonicity fact, which we shall use implicitly.

Lemma 2.7 *If $\widehat{M} \stackrel{w}{\Longrightarrow}$ then $\widehat{M}' \stackrel{w}{\Longrightarrow}$ for any $\widehat{M}' \geq \widehat{M}$.*

Note that every increasing chain of ω-markings has a unique ω-marking as the least upper bound. For a set $\mathcal{M}$ of ω-markings we define its *closure* $Cl(\mathcal{M})$ to be $\mathcal{M}$ enriched by such least upper bounds.

We shall also make use of the next easily derived lemma.

Lemma 2.8 *$\mathcal{T}(\widehat{M}') \subseteq \mathcal{T}(\widehat{M})$ for any ω-markings $\widehat{M}' \leq \widehat{M}$. For a chain $\widehat{M}_1 \leq \widehat{M}_2 \leq \cdots$ with least upper bound $\widehat{M}$, we have*

$$\bigcup_{i\geq 1} \mathcal{T}(\widehat{M}_i) \;=\; \mathcal{T}(\widehat{M}).$$

An important observation is that there can only be finitely many maximal elements in any set of ω-markings. This follows from the next easily derived lemma, known as Dickson's Lemma [2].

Lemma 2.9 *Given an infinite sequence of ω-markings $\widehat{M}_1, \widehat{M}_2, \widehat{M}_3, \ldots$, there are indices $i_1 < i_2 < i_3 < \cdots$ such that $\widehat{M}_{i_1} \leq \widehat{M}_{i_2} \leq \widehat{M}_{i_3} \leq \cdots$.*

Proof: Viewing the ω-markings as mappings from the finite set P to $I\!N_\omega$, the result can be established by a straightforward induction on the cardinality of P. □

Given a marking M of a net N, we can effectively find the maximal elements of $Cl(\mathcal{R}(M))$. This can be achieved by the technique of *coverability trees* [20]; similarly we can get the following.

Lemma 2.10 *Given an ω-marking $\widehat{M}$ and an action symbol $a \in \Sigma$, we can effectively construct the maximal elements of $Cl(\{\widehat{M}' \,:\, \widehat{M} \stackrel{a}{\Longrightarrow} \widehat{M}'\})$.*

We also need an extension of Lemma 2.9 based on Higman's Theorem [5]. Firstly we say that a partially-ordered set $\langle A, \leq \rangle$ has the *finite basis property (fbp)* iff every infinite sequence of elements of A has an infinite (not necessarily strictly) ascending subsequence.

Theorem 2.11 *If $\langle A, \leq \rangle$ has the fbp then so does $\langle A^*, \leqslant \rangle$, where*

$$\leqslant \;=\; \Big\{ \langle\, a_1 a_2 \cdots a_n,\; v_0 b_1 v_1 b_2 \cdots v_{n-1} b_n v_n \,\rangle \;:\; a_i, b_i \in A,\; v_i \in A^*,\; a_i \leq b_i \Big\}.$$

Corollary 2.12 *The set $\mathcal{P}_{\mathit{fin}}(I\!N_\omega^P)$ of all finite sets of ω-markings for a place set P has the fbp with respect to the following ordering:*

$$\mathcal{M} \leq \mathcal{M}' \quad \textit{iff} \quad \textit{for all } \widehat{M} \in \mathcal{M} \textit{ there exists } \widehat{M}' \in \mathcal{M}' \textit{ such that } \widehat{M} \leq \widehat{M}'.$$

Proof: By Lemma 2.9, $\langle I\!N_\omega^P, \leq \rangle$ has the fbp. Hence $\langle (I\!N_\omega^P)^*, \leqslant \rangle$ also has the fbp. The corollary is then clear from the fact that any finite set $\mathcal{M}$ can be viewed as a string of its elements. □

We also need an additional technical result. Let us call a set of markings $\mathcal{M} \subseteq I\!N^P$ *simple* iff there is a disjoint partition $P = P_1 \cup P_2$, a fixed mapping $fix : P_1 \to I\!N$ and a constant n s.t. $\mathcal{M} = \{\, M \mid \forall p \in P_1 : M(p) = fix(p), \forall p \in P_2 : M(p) \geq n \,\}$. The next result shows that it is semidecidable if a simple set $\mathcal{M}_2$ is reachable via markings from a simple set $\mathcal{M}_1$ whose nonfixed values can be arbitrarily large. (In fact, the result can be easily generalized to semilinear sets.)

Lemma 2.13 *Let M_0 be a marking of a net N, and let $\mathcal{M}_1, \mathcal{M}_2$ be simple sets of markings of N; let P_1, P_2 be the partition relevant to $\mathcal{M}_1$. It is semidecidable if for any m there is $M \in \mathcal{M}_1$ s.t. $\forall p \in P_2 : M(p) > m$ and $M_0 \longrightarrow^* M \longrightarrow^* \mathcal{M}_2$.*

Proof: Semidecidability can be established by another appeal to Theorem 2.11. It is similar to the proof of Theorem 6.5 in [9], therefore we shall just state here that the answer to the question is 'yes' iff there are sequences of transitions

$$M_0 \xrightarrow{u_1} M_1 \xrightarrow{u_2} M_2 \xrightarrow{u_3} \cdots \xrightarrow{u_k} M_k \xrightarrow{u_{k+1}} M_{k+1} \xrightarrow{u_{k+2}} \cdots \xrightarrow{u_{k+\ell}} M_{k+\ell}$$

and

$$M_0 \xrightarrow{u_1 w_1} M'_1 \xrightarrow{u_2 w_2} M'_2 \xrightarrow{u_3 w_3} \cdots \xrightarrow{u_k w_k} M'_k \xrightarrow{u_{k+1} w_{k+1}} M'_{k+1} \xrightarrow{u_{k+2} w_{k+2}} \cdots \xrightarrow{u_{k+\ell} w_{k+\ell}} M'_{k+\ell}$$

such that $M_i \le M'_i$ for all $i = 1, 2, \ldots, k+\ell$, $M_k, M'_k \in \mathcal{M}_1$, $M_{k+\ell}, M'_{k+\ell} \in \mathcal{M}_2$, and $M_k(p) < M'_k(p)$ for all $p \in P_2$. (Note that $w_1, w_2, \ldots, w_{k+\ell}$ can be 'pumped' making M'_k larger and larger on P_2.)

The described condition is obviously semidecidable. $\square$

3 Trace equivalence

3.1 Decidability of (strong and weak) trace equivalence

Here we demonstrate the decidability of the following problem:

> Given a marking M_0 of a net N labelled by action set Σ, and a state r_0 of a finite-state LTS R defined over the same action set Σ, is $\mathcal{T}(M_0) = \mathcal{T}(r_0)$?

To do this, we show decidability for the trace inclusion problem in both directions: $\mathcal{T}(M_0) \subseteq \mathcal{T}(r_0)$ and $\mathcal{T}(r_0) \subseteq \mathcal{T}(M_0)$. Without loss of generality we shall suppose that R has no τ labels and is deterministic, i.e., for each state r and each label a there is at most one r' such that $r \xrightarrow{a} r'$; this can be achieved using the standard powerset construction for nondeterministic finite automata (cf. e.g. [7]).

We begin with the first inclusion. In fact, here even the problem $\mathcal{T}(M_0) \subseteq \mathcal{T}(M'_0)$ where M'_0 is a marking of a deterministic net N' is decidable. This can be shown by a reduction to the Petri net reachability problem. Nevertheless we provide an alternative simple proof which does not rely on the decidability of reachability.

Proposition 3.1 $\mathcal{T}(M_0) \subseteq \mathcal{T}(r_0)$ *is decidable.*

Proof: Firstly we can observe the semidecidability of the complementary problem $\mathcal{T}(M_0) \not\subseteq \mathcal{T}(r_0)$. For this, it suffices to generate all sequences from $(\Sigma \setminus \{\tau\})^*$ and to stop when some $w \in \mathcal{T}(M_0) \setminus \mathcal{T}(r_0)$ is found. Decidability of the last condition can be established, for example, by the coverability graph technique; but for our purpose, semidecidability of the problem $w \in \mathcal{T}(M_0)$ suffices, and this is obvious.

Now define the binary relation $S = \left\{ \langle\, \widehat{M}, r \,\rangle \;:\; \mathcal{T}(\widehat{M}) \subseteq \mathcal{T}(r) \right\}$ between ω-markings of the net and states of the LTS. Recalling Lemma 2.8, we can see that S is in fact the downwards closure of the subset of its (finitely many) maximal elements (we put $\langle\, \widehat{M}, r \,\rangle \leq \langle\, \widehat{M}', r' \,\rangle$ iff $\widehat{M} \leq \widehat{M}'$ and $r = r'$). Now observe the following simple fact.

If a set X of pairs $\langle\, \widehat{M}, r \,\rangle$ satisfies the condition

($\star$) For any $\langle\, \widehat{M}, r \,\rangle \in X$ and any $a, \widehat{M}'$ such that $\widehat{M} \xrightarrow{a} \widehat{M}'$ there is r' such that $r \xrightarrow{a} r'$ and $\langle\, \widehat{M}', r' \,\rangle \in X$ (we put $r' = r$ when $a = \tau$).

then $X \subseteq S$.

It is also clear that if X is downwards closed then it suffices to verify ($\star$) only for its maximal elements.

Since S satisfies ($\star$) (recall that for each r and a there is at most one r' such that $r \xrightarrow{a} r'$), to demonstrate $\mathcal{T}(M_0) \subseteq \mathcal{T}(r_0)$ it suffices to generate a (finite) set S' of pairwise incomparable elements $\langle\, \widehat{M}, r \,\rangle$ such that its downwards closure satisfies ($\star$) and contains $\langle\, M_0, r_0 \,\rangle$; this last condition is obviously decidable.

Thus we have demonstrated the semidecidability, and therefore the decidability, of the trace inclusion problem $\mathcal{T}(M_0) \subseteq \mathcal{T}(r_0)$. $\square$

Proposition 3.2 $\mathcal{T}(r_0) \subseteq \mathcal{T}(M_0)$ *is decidable.*

Proof: We describe a terminating algorithm for constructing a tree of the following description. All nodes of the tree are labelled by pairs $\langle\, r, \mathcal{M} \,\rangle$ where r is a state of R and $\mathcal{M}$ is a set of pairwise incomparable ω-markings of N (and hence is finite); a node label $\langle\, r, \mathcal{M} \,\rangle$ is intended to mean

$$\mathcal{T}(r) \;\subseteq\; \bigcup_{\widehat{M} \in \mathcal{M}} \mathcal{T}(\widehat{M}).$$

Observe that if $\langle\, r, \mathcal{M} \,\rangle$ is incorrect with respect to the intended meaning then $\langle\, r, \mathcal{M}' \,\rangle$ is surely incorrect whenever $\mathcal{M}' \leq \mathcal{M}$. The arcs in the tree are labelled from $\Sigma \setminus \{\tau\}$.

The tree is defined inductively as follows.

- The root node is labelled by $\langle\, r_0,\, \{M_0\} \,\rangle$.
- From the node $\langle\, r, \mathcal{M} \,\rangle$, we construct its (finitely many) successors as follows. For each $a \in \Sigma \setminus \{\tau\}$ and r' such that $r \xrightarrow{a} r'$ (r' is unique with respect to a due to the determinism of R), we add the successor $\langle\, r', \mathcal{M}' \,\rangle$ via an arc labelled by a, where $\mathcal{M}'$ is the set of all maximal ω-markings $\widehat{M}'$ for which there is $\widehat{M} \in \mathcal{M}$ such that $\widehat{M} \xRightarrow{a} \widehat{M}'$ (by Lemma 2.10, we can construct the maximal a-successors for each $\widehat{M} \in \mathcal{M}$, and then take the maximal among all of them);

- Any node $\langle\, r, \mathcal{M}\,\rangle$ will be considered as a leaf if
  - either $\mathcal{M} = \emptyset$ (in which case the leaf is deemed to be *unsuccessful*); or
  - either r is a dead state (has no successors) in R, or there is an ancestor $\langle\, r, \mathcal{M}'\,\rangle$ such that $\mathcal{M}' \leq \mathcal{M}$ (in which case the leaf is deemed to be *successful*).

By Corollary 2.12, this tree must be finite, and therefore our algorithm is guaranteed to terminate.

Having constructed the tree, the relevant question can be answered as follows: if there is an unsuccessful leaf $\langle\, r, \emptyset\,\rangle$ then $\mathcal{T}(r_0) \not\subseteq \mathcal{T}(M_0)$; otherwise $\mathcal{T}(r_0) \subseteq \mathcal{T}(M_0)$. For the verification of the correctness, first note that for any node $\langle\, r, \mathcal{M}\,\rangle$ reached from the root by a path labelled by w we have $r_0 \xrightarrow{w} r$, $\mathcal{M}$ being the set of all maximal ω-markings $\widehat{M}$ such that $M_0 \stackrel{w}{\Longrightarrow} \widehat{M}$; therefore $\mathcal{M} = \emptyset$ means $w \notin \mathcal{T}(M_0)$. Hence the correctness in case of an unsuccessful leaf is clear.

Suppose then that all leaves are successful, and in spite of this $\mathcal{T}(r_0) \not\subseteq \mathcal{T}(M_0)$. Choose some $w \in \mathcal{T}(r_0) \backslash \mathcal{T}(M_0)$ of minimal length. It can be written $w = uv$ where u corresponds to a branch in our tree finishing in some leaf $\langle\, r, \mathcal{M}\,\rangle$; then it must hold that $v \in \mathcal{T}(r)$ but $v \notin \mathcal{T}(\widehat{M})$ for each $\widehat{M} \in \mathcal{M}$. This node must have an ancestor $\langle\, r, \mathcal{M}'\,\rangle$ with $\mathcal{M}' \leq \mathcal{M}$; hence we must also have that $v \notin \mathcal{T}(\widehat{M})$ for each $\widehat{M} \in \mathcal{M}'$. This implies that we can write $w = u_1 u_2 v$ (where u_2 is nonempty) in such a way that $u_1 v \in \mathcal{T}(r_0) \setminus \mathcal{T}(M_0)$, which contradicts the minimality of the length of w. □

3.2 Undecidability of strong trace finiteness

In this subsection we demonstrate that it is undecidable whether or not a given τ-free net is trace-equivalent to some (unspecified) finite automaton. In fact, our construction shows that the undecidability result holds for any equivalence which refines the trace equivalence and is refined by simulation equivalence; the construction can also be easily modified to extend the undecidability to ready-simulation equivalence (see, e.g., [3] for definitions; the modification is described in [12]). However, trace equivalence is our only concern here. This undecidability result contrasts with the decidability result for bisimilarity presented in the next section; it also contrasts with the decidability result of Valk and Vidal-Naquet [21] for the regularity of the trace set in the case where the transitions are uniquely labelled.

To demonstrate this result, we rely on the undecidability of the halting problem for Minsky counter machines. To a counter machine C (zero input values are supposed), we construct a net N_C with initial marking M_0 (inspired by [10] as modified in [6]) for which we can demonstrate the following:

1. If the counter machine C halts, then M_0 is trace equivalent to some finite-state process r;
2. If the machine C does not halt, then M_0 is not trace equivalent to any finite-state process r.

Remark. The above mentioned extension of the undecidability result consists in the fact that 'trace equivalence' can be replaced by 'simulation equivalence' (or even 'ready-simulation equivalence' for the modified construction) in case 1 above.

Formally, a *Minsky machine* can be defined as a sequence of labelled instructions

$$\begin{array}{ll} \mathtt{X}_1 & : \mathrm{comm}_1 \\ \mathtt{X}_2 & : \mathrm{comm}_2 \\ & \quad \ldots \\ \mathtt{X}_{n-1} & : \mathrm{comm}_{n-1} \\ \mathtt{X}_n & : \mathtt{halt} \end{array}$$

representing a simple program which uses counters $\mathtt{c}_1$, $\mathtt{c}_2$, ..., $\mathtt{c}_m$, where each of the first $n-1$ instructions is either of the form

$$X \quad : \quad \mathtt{c}_j\mathtt{:=c}_j\mathtt{+1;\ goto}\ X'$$

or of the form

$$\begin{array}{lll} X & : & \mathtt{if\ c}_j\mathtt{=0\ then\ goto}\ X' \\ & & \mathtt{else\ c}_j\mathtt{:=c}_j\mathtt{-1;\ goto}\ X'' \end{array}$$

Here we suppose that a Minsky machine C starts executing with the value 0 in each of the counters and the control at label $\mathtt{X}_1$. When the control is at label $\mathtt{X}_k$ $(1 \le k < n)$, the machine executes instruction comm_k, modifying the contents of the counters and transferring the control to the appropriate label mentioned in the instruction. The machine halts if and when the control reaches the `halt` instruction at label $\mathtt{X}_n$.

We recall now the well-known fact that the halting problem for Minsky Machines is undecidable [18]: there is no algorithm which decides whether or not a given Minsky machine halts. This is true even when restricting attention to two-counter machines only; nevertheless it is technically convenient to consider the general case here.

Given a Minsky machine C, we define the net $N_C = \langle\, P, T, F, \Sigma, \ell \,\rangle$ as follows.

- The set of places is $P = \{\, c_1, c_2, \ldots, c_m, X_1, X_2, \ldots, X_n, U \,\}$. (The initial marking M_0 will consist of just one token, located on the place X_1; and in general, a marking will have a token on some place X_i representing the Minsky machine at that particular instruction label, and some number of tokens on each of the places c_j representing those particular values for the counters.)
- The set of actions labelling the transitions is $\Sigma = \{i, d, z\}$, denoting the machine events *increment*, *decrement*, and *zero*, respectively.
- For every instruction of the form

$$X \quad : \quad \mathtt{c}_j\mathtt{:=c}_j\mathtt{+1;\ goto}\ X'$$

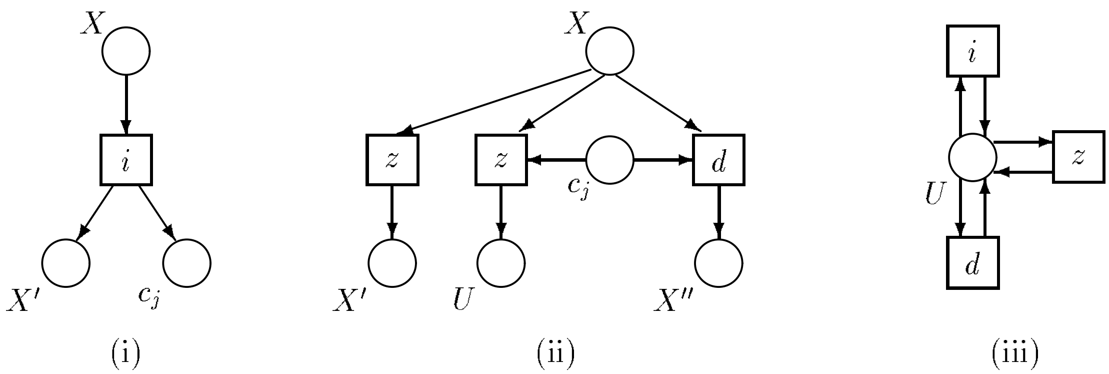

Figure 1: Constructions for N_C

the net has a transition labelled by i with the single input place X and the two output places X' and c_j; see Figure 1(i).

- For every instruction of the form

  X : `if` c_j`=0 then goto` X'
  `else` c_j`:=`c_j`-1; goto` X''

  the net has a transition labelled by d with the two input places X and c_j, and the single output place X''; and two transitions labelled by z, the first with the single input place X and the single output place X', and the second with the two input places X and c_j, and the single output place U; see Figure 1(ii).

- there are three further transitions associated with the place U (for 'universal'). They each have U as both their single input place and their single output place, and they are labelled by i, d, and z, respectively; see Figure 1(iii).

The net N_C simulates the Minsky machine C in a weak sense: there is a unique computation of the net corresponding to the computation of the machine, but there can be 'invalid' transition sequences; these arise due to z-transitions being performed when the relevant counter place c_j is not empty (and the appropriate d-transition is in fact the 'valid' transition); note that invalid z-transitions can lead equally to the universal state from which any action is possible forevermore.

Lemma 3.3 *If C halts then M_0 is trace equivalent to some finite-state process r_0.*

Proof: The backbone of the LTS R containing r_0 will be a (finite) path corresponding to the (valid) computation of C (which halts by assumption); see Figure 2. The initial state of this path will be r_0. Outside of this path there will be one further state u with three 'loops' labelled by i, d and z. From any state on the path which has an outgoing arc labelled by d, we shall have a further arc labelled z leading to the state u.

It is obvious then that $\mathcal{T}(M_0) = \mathcal{T}(r_0)$. □

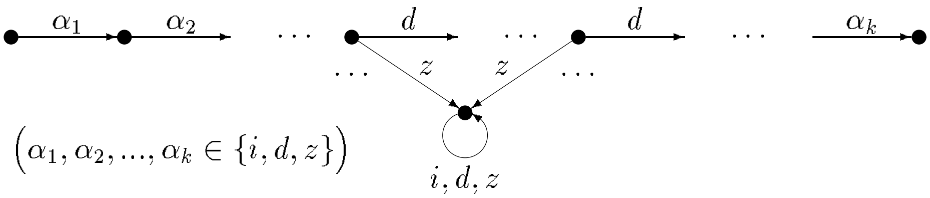

Figure 2: Construction for R

For the opposite direction, we can assume without loss of generality that in any infinite computation of C we can find for any $q \in \mathbb{N}$ a subcomputation during which some counter is decreased q times in succession. This is possible, for example, by including three extra counters a_1, a_2 and a_3, and replacing each original instruction

X_i : comm$_i$

by the sequence of eight instructions

X_i :	`a1:=a1+1; goto` Y_i^1	$*$	*increment* a_1
Y_i^1 :	`if a1=0 then goto` Y_i^3	$*$	<u>*while*</u> $a_1 > 0$ <u>*do*</u>
	`else a1:=a1-1; goto` Y_i^2	$*$	*decrement* a_1
Y_i^2 :	`a2:=a2+1; goto` Y_i^1	$*$	*increment* a_2
Y_i^3 :	`if a2=0 then goto` Y_i^6	$*$	<u>*while*</u> $a_2 > 0$ <u>*do*</u>
	`else a2:=a2-1; goto` Y_i^4	$*$	*decrement* a_2
Y_i^4 :	`a1:=a1+1; goto` Y_i^5	$*$	*increment* a_1
Y_i^5 :	`a3:=a3+1; goto` Y_i^3	$*$	*increment* a_3
Y_i^6 :	`if a3=0 then goto` Y_i^7	$*$	<u>*while*</u> $a_3 > 0$ <u>*do*</u>
	`else a3:=a3-1; goto` Y_i^6	$*$	*decrement* a_3
Y_i^7 :	comm$_i$		

The effect of this transformation is to maintain in counter a_1 the number of commands executed by the Minsky machine, and before executing each command to cause the counter a_3 to be set to this value and then to be repeatedly decremented down to 0; this clearly leads to longer and longer sequences of decrement actions, without changing the (non-)halting behaviour of the original program.

Lemma 3.4 *If C does not halt then $\mathcal{T}(M_0)$ is different from the trace set of any finite-state process r_0.*

Proof: Suppose that $\mathcal{T}(M_0) = \mathcal{T}(r_0)$ for some finite-state process r_0 taken from a q-state LTS R. Then r_0 also must allow the prefix of a 'valid' computation sequence which includes a contiguous sequence of q decrement actions. Using the Pumping Lemma for finite-state machines [7], this means that r_0 must be able to reach a state by following a valid computation sequence from which it can follow an arbitrary number of decrement actions, which clearly is not possible for M_0. Hence $\mathcal{T}(M_0) \neq \mathcal{T}(r_0)$ which contradicts our assumption. □

Based on the two lemmas and the undecidability of the halting problem for Minsky machines, we can derive our undecidability result.

Theorem 3.5 *It is undecidable whether or not a given τ-free net is trace equivalent to some (unspecified) finite-state LTS.*

4 Bisimulation equivalence

4.1 Decidability of strong bisimulation equivalence

The proof is based on the general method described in Section 2. Given a marking M_0 of a τ-free net N and a state r_0 of an n-state LTS R, the question $M_0 \sim_n r_0$ is obviously decidable (cf. Proposition 2.2). Therefore by Corollary 2.5, it suffices to show the decidability of the question as to whether the set

$$\textsc{Inc} \;=\; \textsc{Inc}_n^R \,\cap\, \{\, M \;:\; M \text{ is a marking of } N \,\}$$

is reachable from M_0.

We say that a marking L of a net N is *n-bounded* iff $L(p) \leq n$ for each place p. For every n-bounded marking L, we define $L^{\geq n}$ as the set of all markings M such that $L(p) = \min\big(n,\, M(p)\big)$ for each place p. Note that for every marking M there is a unique n-bounded marking L_M such that $M \in L_M^{\geq n}$; that is, $M \in L^{\geq n}$ iff $L = L_M$. Also, there are only finitely many n-bounded markings; and $M \sim_n L_M$, so $M \in \textsc{Inc}$ iff $L_M^{\geq n} \subseteq \textsc{Inc}$. Hence, $\textsc{Inc}$ is an effectively constructible union

$$\textsc{Inc} \;=\; L_1^{\geq n} \cup L_2^{\geq n} \cup \cdots \cup L_k^{\geq n}$$

for some n-bounded markings $L_1, L_2, \ldots, L_k$.

Theorem 4.1 *The problem $M_0 \sim r_0$ is decidable.*

Proof: From the above considerations it follows that it suffices to show decidability of the following problem:

> Given an n-bounded marking L, is the set $L^{\geq n}$ reachable; that is, is there some $M \in L^{\geq n}$ such that $M_0 \longrightarrow^* M$?

But this problem is easily reducible to the reachability problem: for each place p such that $L(p) = n$ we can add an extra transition which just removes a token from p, and then ask if L is reachable. □

4.2 Decidability of strong bisimulation finiteness

We now prove that it is decidable whether or not a given marking M_0 of a given τ-free net N is bisimilar to some (unspecified) finite-state process. We shall refer to this problem as the *strong bisimulation finiteness problem*, called *strong b-finiteness problem* for short.

A marking M is infinite with respect to strong bisimilarity (*b-infinite* for short) iff there exist infinitely many markings $M_1, M_2, M_3, \ldots$ reachable from M such that $M_i \not\sim M_j$ for $i \neq j$. Since the strong equivalence problem is decidable, the strong b-finiteness problem is obviously semidecidable; it suffices to generate all finite-state processes r_0 and to check if $M_0 \sim r_0$. Therefore, it suffices to show that b-infiniteness is semidecidable.

This semidecidability can be informally explained as follows. A sufficient condition for the existence of infinitely many nonbisimilar reachable states is that there are infinitely many reachable states with different 'distances' to a certain '(n-step) behaviour'. It turns out that in our case it is also a necessary condition and, in addition, it is semidecidable. This imprecise idea is formalized in the following.

We fix a labelled Petri net $N = \langle P, T, F, \Sigma, \ell \rangle$ and introduce some notation. Let $P = P_1 \cup P_2$ where P_1, P_2 are disjoint and $P_2 \neq \emptyset$. For mappings $M_1 : P_1 \longrightarrow I\!N$ and $M_2 : P_2 \longrightarrow I\!N$, (M_1, M_2) denotes the marking of N whose projection onto P_1 is M_1 while the projection onto P_2 is M_2. We say 'a marking (M_1, M_2) of N' instead of 'a partition P_1, $P_2 \neq \emptyset$ of P and mappings $M_1 : P_1 \longrightarrow I\!N$, $M_2 : P_2 \longrightarrow I\!N$'. In addition, by $(M, -)$ we mean that there is a partition $P = P_1 \cup P_2$ as above but $(M, -)$ is considered as a marking $(M : P_1 \longrightarrow I\!N)$ of the *subnet* of N obtained by removing all places from P_2, together with their adjacent arcs (which is behaviourally equivalent to putting ω's on places from P_2). Observe that, for any $i \geq 0$, if $M'(p) \geq i$ for each place p then $(M, M') \sim_i (M, -)$.

Lemma 4.2 *If $(M, M_1) \sim (M, M_2) \sim (M, M_3) \sim \cdots$ and $M_1 < M_2 < M_3 < \cdots$ (where $<$ is defined pointwise) then $(M, M_1) \sim (M, -)$.*

Proof: For every $i \geq 0$ there surely exists an index j such that $M_j(p) \geq i$ for each p. Then $(M, -) \sim_i (M, M_j)$ holds and, since $(M, M_1) \sim (M, M_j)$, we also have $(M, -) \sim_i (M, M_1)$. Therefore, $(M, -) \sim_i (M, M_1)$ for every $i \geq 0$, and so $(M, -) \sim (M, M_1)$. □

Lemma 4.3 *A marking M_0 is b-infinite iff there exists a marking $(M, -)$ satisfying one of the following two conditions:*

(1) *$(M, -)$ is b-infinite and there exists a chain $M_1 < M_2 < M_3 < \cdots$ such that $(M, M_i) \in \mathcal{R}(M_0)$ for every $i \geq 1$; or*

(2) *$(M, -)$ is b-finite and there exists a chain $M_1 < M_2 < M_3 < \cdots$ such that $(M, M_i) \in \mathcal{R}(M_0)$ and $(M, M_i) \not\sim (M, -)$ for every $i \geq 1$.*

Proof: $(\Rightarrow)$: If M_0 is b-infinite, then there exists an infinite set of pairwise non-bisimilar reachable markings. Consider any infinite sequence of such markings. By

Lemma 2.9, there is (a certain partition $P = P_1 \cup P_2$, $P_2 \neq \emptyset$, and) an infinite subsequence $(M, M_1), (M, M_2), (M, M_3), \ldots$ such that $M_1 < M_2 < M_3 < \cdots$. So either (1) or (2) holds, according to whether $(M, -)$ is b-finite or b-infinite.

($\Leftarrow$): Let $\mathcal{M} = \{(M, M_i) \ : \ i \geq 1\}$. If $\mathcal{M}$ contains infinitely many pairwise non-bisimilar markings, then M_0 is b-infinite, and we are done. So assume that $\mathcal{M}$ contains infinitely many pairwise bisimilar markings. By Lemma 4.2, all of these markings must be bisimilar to $(M, -)$, and so (2) cannot hold. Thus (1) must hold, meaning that $(M, -)$ is b-infinite. Therefore M_0 must itself be b-infinite. □

Theorem 4.4 *It is decidable whether or not a marking M_0 of a net N is b-finite.*

Proof: We proceed by induction on the number of places of N. If N has no places, then $M_0 = \emptyset$, and is clearly b-finite. Assume now that N has some places. As already mentioned, it suffices to show semidecidability of the b-infiniteness problem. To this aim, it obviously suffices to show semidecidability of conditions (1) and (2) of Lemma 4.3.

For this purpose, we enumerate all markings $(M, -)$ of N for all partitions P_1, P_2 such that $P_2 \neq \emptyset$. Given a marking $(M, -)$, we can decide by the induction hypothesis if it is b-finite or b-infinite; moreover:

(1) The existence of a chain $M_1 < M_2 < M_3 < \cdots$ such that $(M, M_i) \in \mathcal{R}(M_0)$ for every $i \geq 1$ is surely semidecidable: just put $\mathcal{M}_1 = \mathcal{M}_2 = \{\ (M, M') \mid M' \text{ is arbitrary }\}$ and apply Lemma 2.13.

(2) If $(M, -)$ is b-finite, then the existence of a chain $M_1 < M_2 < M_3 < \cdots$ such that $(M, M_i) \in \mathcal{R}(M_0)$ and $(M, M_i) \not\sim (M, -)$ for every $i \geq 1$ is also semidecidable: if $(M, -)$ is b-finite then $(M, -) \sim r$ for a state r of a finite-state LTS R; we can suppose a concrete R (in fact, in our method we already have it when establishing that $(M, -)$ is b-finite); let n denote the number of states of R.

   We say that a chain $M_1 < M_2 < M_3 < \cdots$ is *adequate* if it satisfies the conditions of (2).

   Claim. There exists an adequate chain iff there exists an n-bounded marking L of N satisfying the following two conditions

   (a) $L \in \textsc{Inc}$; and

   (b) there exists a chain $M_1 < M_2 < M_3 < \cdots$ and markings $M'_1, M'_2, M'_3, \ldots \in L^{\geq n}$ such that $M_0 \longrightarrow (M, M_i) \longrightarrow M'_i$ for every $i \geq 1$. (Recall that the domain of M_0 and M'_i is P while it is P_1 for M and P_2 for M_i.)

   Proof of the Claim.

   ($\Rightarrow$): Let $M_1 < M_2 < M_3 < \cdots$ be an adequate chain. There exists an index i_0 such that $M_i \geq (n, n, \ldots, n)$ for every $i \geq i_0$. For $i \geq i_0$ we have $(M, M_i) \not\sim (M, -)$ by assumption (and so $(M, M_i) \not\sim r$), but $(M, M_i) \sim_n (M, -)$ (and so $(M, M_i) \sim_n r$). Recalling Section 4.1, there exists an n-bounded marking $L_i \in \textsc{Inc}$ such that $(M, M_i) \longrightarrow^* L_i^{\geq n}$.

By the pigeonhole principle there exists an n-bounded marking L and infinitely many indices $i_1 < i_2 < i_3 < \cdots$ such that $L = L_{i_1} = L_{i_2} = L_{i_3} = \ldots$. Clearly, L satisfies (a) and the subchain $M_{i_1} < M_{i_2} < M_{i_3} < \cdots$ satisfies (b).

($\Leftarrow$): Let M_i be an arbitrary marking of the chain given by (b). We prove $(M, M_i) \not\sim (M, -)$, which shows that the chain is adequate. Since $M_0 \longrightarrow (M, M_i) \longrightarrow M_i'$ for some marking $M_i' \in L^{\geq n}$, we have $(M, M_i) \longrightarrow^* L^{\geq n}$. By (a) and Section 4.1 we have $(M, M_i) \not\sim r$, which together with $(M, -) \sim r$ implies $(M, M_i) \not\sim (M, -)$.

It remains to prove the semidecidability of conditions (a) and (b) for a given n-bounded marking L. Condition (a) is clearly decidable. For condition (b), put $\mathcal{M}_1 = \{ (M, M') \mid M' \text{ is arbitrary} \}$, $\mathcal{M}_2 = L^{\geq n}$ and apply Lemma 2.13. □

4.3 Undecidability of weak bisimulation equivalence

We next show that the question $M_0 \approx r_0$ is undecidable. In fact, we prove that neither of the problems $M_0 \approx r_0$ and $M_0 \not\approx r_0$ is semidecidable. From the proof of this result, we actually get a fixed 7-state transition system R_{fix} with a distinguished state r_{fix} such that $M_0 \approx r_{\text{fix}}$ is undecidable. In fact, even $M_0 \approx_4 r_{\text{fix}}$ is undecidable.

As the basis for our reduction, we use the following undecidable problem from Petri net theory:

Containment problem: Given two Petri nets N_1 and N_2 defined over the same set of places and initial marking M, is $\mathcal{R}_{N_1}(M) \subseteq \mathcal{R}_{N_2}(M)$?

(To avoid the obvious confusion, we write $\mathcal{R}_{N_1}(M)$ and $\mathcal{R}_{N_2}(M)$ rather than $\mathcal{R}(M)$ so as to indicate the underlying net.) The undecidability of this problem was first demonstrated by Rabin (see [4]) by means of a reduction from Hilbert's 10th problem. A reduction from the halting problem for Minsky machines can be found in [10]. In the next section we shall need to describe the latter reduction in more detail.

Let two Petri nets $N_1 = (P, \Sigma, T_1, F_1, \ell_1)$ and $N_2 = (P, \Sigma, T_2, F_2, \ell_2)$ be given, along with a common initial marking M. Without loss of generality, we shall assume that $|\mathcal{R}_{N_2}(M)| \geq 2$ and that $\mathbf{0} \notin \mathcal{R}_{N_1}(M) \cup \mathcal{R}_{N_2}(M)$. (By $\mathbf{0}$, we mean the marking which associates the value 0 to each place.) We shall describe a construction of a new net N with initial marking M_0 such that

1. if $\mathcal{R}_{N_1}(M) \not\subseteq \mathcal{R}_{N_2}(M)$ then $M_0 \approx r_1$, where r_1 is taken from the finite transition system R shown in Figure 3; and
2. if $\mathcal{R}_{N_1}(M) \subseteq \mathcal{R}_{N_2}(M)$ then $M_0 \approx r_5$, where r_5 is again taken from R.

(The state r_0 of R is used in the next section.)

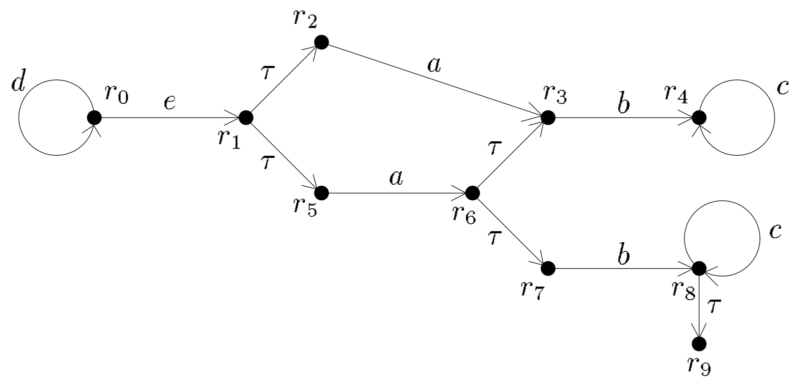

Figure 3: The Finite-state system R

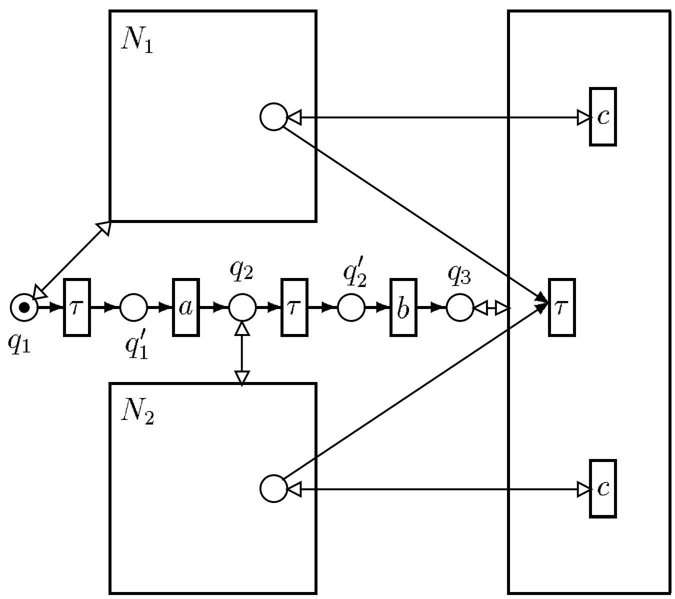

Figure 4: Constructing the net N from N_1 and N_2

When defining N we use the following notion. A place p is a *run-place* of a set T of transitions if $\langle p, t \rangle$ and $\langle t, p \rangle$ are both arcs for every $t \in T$. In particular, the transitions of T can occur only when p holds at least one token.

Figure 4 shows a schema of the net N. To construct it, we first take the disjoint union of N_1 and N_2, relabelling all transitions by τ. We assume that the places of N_i (for $i = 1, 2$) are given by $P_i = \{p_i \ : \ p \in P\}$. As a part of the initial marking M_0, we put M on N_1 and on N_2.

We then add further places and transitions as indicated. The place q_1 is a run-place of T_1 (graphically represented by a double pointed white arrow), and contains initially one token. This token can be moved by a τ-transition to a place q_1', and then by an a-transition to q_2, which is a run-place of T_2. From q_2, the token can be moved by another τ-transition to q_2' and by a b-transition to q_3, which is a run-place of an additional set of transitions. This set contains:

- a τ-transition for every pair $\langle p_1, p_2 \rangle$ $(p \in P)$; the transition has p_1 and p_2 as input places, and no output place; when it occurs, it simultaneously decreases the marking of p_1 and p_2,; and
- a c-transition for each place p_i of N_1 and N_2; the transition has p_i as the unique input and output place.

We denote a marking of N as a vector with three components: the first and third components are the projections of the marking onto N_1 and N_2, respectively, while the second indicates which place of the set $\{q_1, q_1', q_2, q_2', q_3\}$ currently holds a token. The initial marking M_0 of N is (M, q_1, M).

From this initial marking M_0, the net N can execute τ-transitions corresponding to the transitions of N_1. If at some moment the τ-transition occurs taking the q_1 token to q_1', then a marking (M_1, q_1', M) is reached, the submarking M_1 becomes 'frozen', and the only available transition is the a-transition leading to the marking (M_1, q_2, M). From here, N can then execute τ-transitions corresponding to the transitions of N_2. Again, if at some moment the τ-transition occurs taking the q_2 token to q_2', then a marking (M_1, q_2', M_2) is reached, and the submarking M_2 becomes 'frozen' as well.

The following Proposition is then easy to prove.

Proposition 4.5

1. *If* $\mathcal{R}_{N_1}(M) \subseteq \mathcal{R}_{N_2}(M)$ *then* $M_0 \approx r_5$.
2. *If* $\mathcal{R}_{N_1}(M) \not\subseteq \mathcal{R}_{N_2}(M)$ *then* $M_0 \approx r_1$.

Proof: A bisimulation containing the pair $\langle M_0, r_5 \rangle$ if $\mathcal{R}_{N_1}(M) \subseteq \mathcal{R}_{N_2}(M)$ and the pair $\langle M_0, r_1 \rangle$ if $\mathcal{R}_{N_1}(M) \not\subseteq \mathcal{R}_{N_2}(M)$ consists of the following pairs:

$$\begin{array}{lll}
\langle\,(M_1,q_1,M_2)\;,\;r_1\,\rangle & \text{where} & \mathcal{R}_{N_1}(M_1)\not\subseteq\mathcal{R}_{N_2}(M_2),\;\mathcal{R}_{N_1}(M_1)\cap\mathcal{R}_{N_2}(M_2)\neq\emptyset;\\
\langle\,(M_1,q_1,M_2)\;,\;r_2\,\rangle & \text{where} & \mathcal{R}_{N_1}(M_1)\cap\mathcal{R}_{N_2}(M_2)=\emptyset;\\
\langle\,(M_1,q_1,M_2)\;,\;r_5\,\rangle & \text{where} & \mathcal{R}_{N_1}(M_1)\subseteq\mathcal{R}_{N_2}(M_2);\\
\\
\langle\,(M_1,q_1',M_2)\;,\;r_2\,\rangle & \text{where} & M_1\notin\mathcal{R}_{N_2}(M_2);\\
\langle\,(M_1,q_1',M_2)\;,\;r_5\,\rangle & \text{where} & M_1\in\mathcal{R}_{N_2}(M_2);\\
\\
\langle\,(M_1,q_2,M_2)\;,\;r_3\,\rangle & \text{where} & M_1\notin\mathcal{R}_{N_2}(M_2);\\
\langle\,(M_1,q_2,M_2)\;,\;r_6\,\rangle & \text{where} & M_1\in\mathcal{R}_{N_2}(M_2)\neq\{M_1\};\\
\langle\,(M_1,q_2,M_2)\;,\;r_7\,\rangle & \text{where} & \mathcal{R}_{N_2}(M_2)=\{M_1\};\\
\\
\langle\,(M_1,q_2',M_2)\;,\;r_3\,\rangle & \text{where} & M_1\neq M_2;\\
\langle\,(M_1,q_2',M_1)\;,\;r_7\,\rangle; & & \\
\\
\langle\,(M_1,q_3,M_2)\;,\;r_4\,\rangle & \text{where} & M_1\neq M_2;\\
\langle\,(M_1,q_3,M_1)\;,\;r_8\,\rangle & \text{where} & M_1\neq\mathbf{0};\\
\langle\,(\mathbf{0},q_3,\mathbf{0})\;,\;r_9\,\rangle. & &
\end{array}$$

This is readily seen to be a bisimulation. □

Theorem 4.6 *Neither the weak equivalence problem $M \approx r$ nor the weak non-equivalence problem $M \not\approx r$ are semidecidable.*

Proof: This follows from the undecidability of the containment problem, using Proposition 4.5 and the fact that $r_1 \not\approx r_5$. □

Thus the problem $N \approx r_5$ is undecidable. More than this, however, we may observe in the above proof that $r_1 \not\approx_4 r_5$; hence also $N \approx_4 r_5$ is undecidable. The 7-state transition system R_{fix} promised at the beginning of the section is obtained by removing r_0, r_1 and r_2 from R, together with their adjacent arcs.

4.4 Undecidability of weak bisimulation finiteness

In this section we demonstrate the undecidability of the weak b-finiteness problem, i.e., given M_0, for some labelled N, is there a state r_0 of a finite-state LTS R such that $M_0 \approx r_0$? To do this, we again use the halting problem for Minsky counter machines; now it is convenient to recall that it is undecidable even when restricted to 2 counters and zero inputs.

As already mentioned, we rely on a reduction from [10]. For our aims here, it is sufficient to recall that there is an algorithm specified as follows:

Input: a 2-counter machine C.

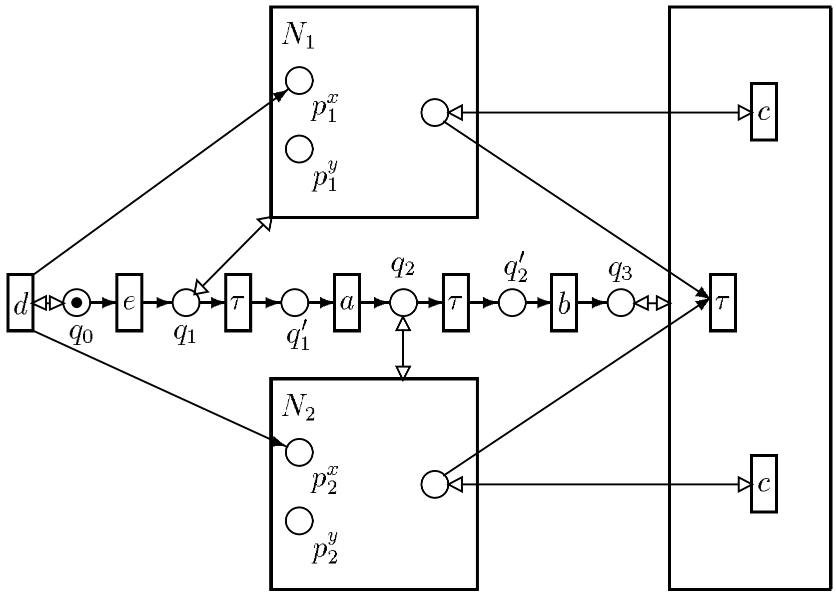

Figure 5: Constructing the net N' from N

Output: two nets N_1 and N_2 defined over the same set of places P including two distinguished places p^x and p^y, and initial marking M. (In fact, N_1, N_2 are almost identical, differing only in that N_1 has an additional transition which is not present in N_2.) These two nets satisfy the following property: if $M^{x,y}$ denotes the marking which differs from M only in the places p^x and p^y where the values are x and y, respectively, then for every $x, y \geq 0$:

$$C \text{ halts on the input } (x,y) \quad \text{iff} \quad \mathcal{R}_{N_1}(M^{x,y}) \not\subseteq \mathcal{R}_{N_2}(M^{x,y}).$$

Now let C be an arbitrary 2-counter machine. We construct another 2-counter machine C' which for input $(x,0)$ runs as follows: first, it checks if $x = 2^k$ for some $k \geq 0$; if this is the case, then it sets the counters to 0 and simulates C, otherwise it halts. We thus have:

- if C halts on input $(0,0)$, then C' halts on every input $(x,0)$, $x \geq 0$;
- if C does not halt on input $(0,0)$ then C' halts on input $(x,0)$ iff x is not a power of 2.

For C', we can construct the above described nets N_1 and N_2. To these, we apply the prior construction depicted in Figure 4; thus we get a net N with a predefined initial marking M_0. We modify this net in the following way (depicted in Figure 5). First, we remove the token from q_1. Second, we add the following new places and transitions:

- a place q_0, initially marked with one token;
- a d-transition with q_0 as the only input place, and q_0, p_1^x and p_2^x as the output places (i.e., q_0 is a run-place for this transition);

- an e-transition, with q_0 as input place and q_1 as output place.

Let N' be the result of this final modification. From its initial marking M'_0, N' can repeatedly execute the d-transition, through which it puts an arbitrary number of tokens x on the places p_1^x and p_2^x. Then, it may execute the e-transition. After that, the place q_1 holds a token, and N' behaves like the net we would obtain by applying the construction of the last section to the nets N_1 and N_2 with initial marking $M^{x,0}$.

Proposition 4.7 *M'_0 is weakly b-finite iff the counter machine C halts on input $(0,0)$.*

Proof: ($\Rightarrow$): If C does not halt on input $(0,0)$, then C' halts on input $(x,0)$ iff x is not a power of 2. Therefore, $\mathcal{R}_{N_1}(M^{x,0}) \subseteq \mathcal{R}_{N_2}(M^{x,0})$ iff $x = 2^k$ for some $k \geq 0$.

Let R be the finite-state transition system of Figure 3. For any x, given the unique marking M reached after executing the d-transition x times in N', we have the following:

(1) if x is not a power of 2, then $M \stackrel{e}{\Longrightarrow} M'$ for some $M' \approx r_1$;

(2) if x is a power of 2, then there is no such M'.

We prove by contradiction that M'_0 is weakly b-infinite. Assume that $M'_0 \approx r'_0$ where r'_0 is a state in some n-state LTS R'. Let r' be a state such that $r'_0 \stackrel{u}{\longrightarrow} r'$, where u is a sequence of actions whose projection onto the set of observable actions is d^{2^n}. By the pumping lemma, $r'_0 \xrightarrow{vw^ix} r'$ for sequences v, w, x and for every $i \geq 0$, where $u = vwx$ and the projection of w onto the set of observable actions is a nonempty sequence of d's. By (1) and (2) we have that there is r'' in R' such that $r' \stackrel{e}{\Longrightarrow} r''$ and $r'' \approx r_1$, and at the same time there is no such r''—a contradiction.

($\Leftarrow$): If C halts on input $(0,0)$, then C' halts for every input $(x,0)$, $x \geq 0$. Therefore after the occurrence of the e-transition we always have $\mathcal{R}_{N_1}(M^{x,0}) \not\subseteq \mathcal{R}_{N_2}(M^{x,0})$, regardless of the value of x. Hence it is clear that $M'_0 \approx r_0$, so M'_0 is weakly b-finite. $\square$

Theorem 4.8 *Neither the weak b-finiteness problem nor the weak b-infiniteness problem is semidecidable.*

Proof: By Proposition 4.7, C does not halt on input $(0,0)$ iff M'_0 is weakly b-infinite. So the weak b-infiniteness problem is not semidecidable. We can also change C' in the following way: if x is not a power of 2, then C' enters an infinite loop. In this case, C does not halt on input $(0,0)$ iff the net M'_0 is weakly b-finite. So the weak b-finiteness problem is not semidecidable either. $\square$

References

[1] Christensen, S., Y. Hirshfeld and F. Moller. Bisimulation is decidable for basic parallel processes. In E. Best (editor), *Proceedings of CONCUR'93: Concurrency Theory, Lecture Notes in Computer Science* **715**:143–157, Springer-Verlag, 1993.

[2] Dickson, L.E. Finiteness of the odd perfect and primitive abundant numbers with distinct factors. *American Journal of Mathematics* **35**:413–422, 1913.

[3] van Glabbeek, R.J. The linear time – branching time spectrum. In J.C.M. Baeten and J.W. Klop (editors), *Proceedings of CONCUR'90: Concurrency Theory, Lecture Notes in Computer Science* **458**, pp278–297, Springer-Verlag, 1990.

[4] Hack, M. The equality problem for vector addition systems is undecidable, *Theoretical Computer Science* **2**:77–95, 1976.

[5] Higman, H. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society* **3**(2), pp326–336, 1952.

[6] Hirshfeld, Y. Petri nets and the equivalence problem. In E. Börger, Y. Gurevich and K. Meinke (editors), *Proceedings of CSL'93: Computer Science Logic, Lecture Notes in Computer Science* **832**:165–174, Springer-Verlag, 1994.

[7] Hopcroft, J.E. and J.D. Ullman. **Introduction to Automata Theory, Languages, and Computation**. Addison Wesley, 1979.

[8] Hüttel, H. Undecidable equivalences for basic parallel processes. In R.K. Shyamasundar (editor), *Proceedings of FSTTCS'93: Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science* **761**, Springer-Verlag, 1993.

[9] Jančar, P. Decidability of a temporal logic problem for Petri nets, *Theoretical Computer Science* **74** (1990) 71–93.

[10] Jančar, P. Undecidability of bisimilarity for Petri nets and some related problems. *Journal of Theoretical Computer Science* **148**, pp281–301, 1995.

[11] Jančar, P. All action based behavioural equivalences are undecidable for labelled Petri nets. *Bulletin of the EATCS* **56**:86–88, 1995.

[12] Jančar, P. Decidability questions for equivalences on Petri nets. Habilitation Thesis. Masaryk University, Brno, Czech Republic (1995)

[13] Jančar, P. and J. Esparza. Deciding finiteness of Petri nets up to bisimulation. In F. Meyer auf der Heide and B. Monien (editors), *Proceedings of ICALP'96: Automata, Languages and Programming, Lecture Notes in Computer Science* **1099**:478–489, Springer-Verlag, 1996.

[14] Jančar, P. and A. Kučera. Bisimilarity of processes with finite-state systems. In F. Moller (editor), *Proceedings of Infinity'97: the 2nd International Workshop on Verification of Infinite State Systems*, *Electronic Notes in Theoretical Computer Science* **9**, URL: `http://www.elsevier.nl/locate/entcs/volume9.html`, 1997.

[15] Jančar, P. and F. Moller. Checking regular properties of Petri nets. In I. Lee and S.A. Smolka (editors), *Proceedings of CONCUR'95: Concurrency Theory*, *Lecture Notes in Computer Science* **962**:348–362, Springer-Verlag, 1995.

[16] Mayr, E. An algorithm for the general Petri net reachability problem. *SIAM Journal of Computing* **13**, pp441–460, 1984.

[17] Milner, R. **Communication and Concurrency**. Prentice Hall, 1989.

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

[19] Olderog, E.-R. **Nets, Terms and Formulas**. Cambridge University Press, 1991.

[20] Peterson, J.L. **Petri Net Theory and the Modeling of Systems**. Prentice Hall, 1981.

[21] Valk, R. and G. Vidal-Naquet. Petri nets and regular languages. *Journal of Computer and System Sciences* **23**(3), pp299–325, 1981.