Journal of Computer and System Sciences **59**, 476–503 (1999)
Article ID jcss.1999.1643, available online at http://www.idealibrary.com on IDEAL®

Petri Nets and Regular Processes

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

Javier Esparza[2]

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

and

Faron Moller[3]

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

Received June 5, 1998; revised April 14, 1999

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 weak versions. (In the weak version a special τ action—likened to an ε-move in automata theory—is considered to be nonobservable.) 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, as well as 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

[1] Partially supported by the Grant Agency of the Czech Republic Grant 201/97/0456 and by a grant from the Swedish STINT Fellowship programme.

[2] Partially supported by the Sonderforschungsbereich 342 of the DFG.

[3] Partially supported by TFR Grant 221-98-103. Corresponding author. E-mail: fm@csd.uu.se.

0022-0000/99 $30.00

behaviour. One of the first questions to ask then for the purpose of (automatic) verification is: (to what extent) is the equivalence decidable?

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 [22] and Olderog [21] and by the "Advances in Petri Nets" volumes of the series Lecture Notes in Computer Science. 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. (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 role to ε-moves in finite automata. The firing of these transitions is assumed to be unobservable.)

Unfortunately, already the strong versions (along with the strong versions of all "reasonable" behavioural equivalences) are undecidable for general Petri nets [10–12], 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 realized 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, that is, 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. 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 finiteness questions are decidable for strong bisimilarity, yet undecidable for weak bisimilarity.

Our results extend and complement previous results by Valk and Vidal-Naquet [23] on the finiteness question for trace equivalence, which they referred to as the *regularity* question, as they were only interested in deciding if the traces describe regular languages. They showed that the regularity of the terminal language of a net—that is, 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.

These problems can be addressed with respect to any semantic equivalence, for example, for any of the observation-based equivalences catalogued by van Glabbeek [3], or any of a variety of non-interleaving semantic equivalences proposed for Petri nets. We restrict our present study to two of the most important observation-based equivalences which happen to lie at opposite ends (with respect to distinguishing power) of van Glabbeek's spectrum.

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 the equivalences which we study. We also present a catalogue of technical results—both old and new—which we 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 the 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 showing 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. The proof is carried out by reduction from the halting problem for Minsky machines.

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 the weak case. 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 [17, 14].

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: $\mathbb{N} = \{0, 1, 2, ...\}$. For a set A, A^* denotes the set of finite sequences of elements of A; the empty sequence is denoted by $\varepsilon \in A^*$. For $u \in A^*$ and $k \in \mathbb{N}$, we denote by u^k the k-fold concatenation of u; and by $|u|$ we denote the length of u.

2.1. Labelled Transition Systems and Equivalences

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

We say that a set of states $S \subseteq \mathscr{S}$ is *reachable* from E, written $E \rightarrow^* S$, iff $E \rightarrow^* F$ for some $F \in S$. The *reachability set* for a state E of $\mathscr{L}$ is defined by $\mathscr{R}_{\mathscr{L}}(E) = \{F : E \rightarrow^* F\}$. (We generally omit the subscript $\mathscr{L}$ when the underlying LTS is clear from the context.)

An LTS $\mathscr{L} = \langle \mathscr{S}, \Sigma, \{\xrightarrow{a}\}_{a \in \Sigma} \rangle$ is *finite-state* iff $\mathscr{S}$ is finite. $\mathscr{L}$ is *image finite* iff $\mathrm{succ}_a(E) = \{F : E \xrightarrow{a} F\}$ is finite for every $E \in \mathscr{S}$ and every $a \in \Sigma$. By a *process* E we refer to a state in a transition system; when necessary, we denote the underlying transition system by $\mathscr{L}(E)$. By referring to a *finite-state process* E, we mean that

$\mathscr{L}(E)$ is finite; a similar convention holds for an *image finite process*. We 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 $\mathscr{B}$ between processes is a *strong bisimulation*, provided that whenever $\langle E, F\rangle \in \mathscr{B}$ for each $a \in \Sigma$,

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

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

- $E \sim_0 F$ for all processes E and F; and
- $E \sim_{n+1} F$ iff for each $a \in \Sigma$:

  — if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $E' \sim_n F'$;

  — 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 (by induction on n). The next two propositions are also easily confirmed folklore.

PROPOSITION 2.1. *For image finite processes E and F, $E \sim F$ iff $E \sim_n F$ for all $n \geqslant 0$.*

Proof. The forward implication can be proved by induction on n; the reverse implication is proved by demonstrating that the relation $\mathscr{B} = \{\langle E, F\rangle : E \sim_n F$ for all $n\}$ is a strong bisimulation. ∎

Let us call $\mathscr{L} = \langle \mathscr{S}, \Sigma, \{\xrightarrow{a}\}_{a \in \Sigma}\rangle$ an *admissible system* iff the state set $\mathscr{S}$ is finite or countably infinite (identified with a set of sequences over a finite alphabet), $\mathscr{L}$ is image finite, and all of the successor functions $\mathrm{succ}_a : \mathscr{S} \to 2^{\mathscr{S}}$ 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.*

Proof. To decide $E \sim_n F$, we need simply resort to the definition of the relations $\sim_n$. $E \not\sim F$ can then be confirmed by deciding each $E \sim_n F$ for $n = 0, 1, 2, \ldots$ until we discover that $E \not\sim_n F$ for some n. ∎

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 ε-moves, as we want to be able to distinguish, for example, between $\xrightarrow{a}$ and $\xrightarrow{\tau a}$; whereas $\varepsilon a = a$, $\tau a \neq a$.)

Given any $a \in \Sigma$ with $a \neq \tau$, we let $E \overset{a}{\Longrightarrow} F$ represent $E \overset{u}{\longrightarrow} F$ for some $u = \tau^k a \tau^\ell$ $(k, \ell \geqslant 0)$; that is, $\overset{a}{\Longrightarrow} = (\overset{\tau}{\longrightarrow})^* \overset{a}{\longrightarrow} (\overset{\tau}{\longrightarrow})^*$. We then let $E \overset{\tau}{\longrightarrow} F$ represent $E \overset{u}{\longrightarrow} F$ for some $u = \tau^k$ $(k \geqslant 0)$. Note that we allow $u = \varepsilon$ so, for example, $E \overset{\tau}{\Longrightarrow} E$ for all E.[4] The relations $E \overset{u}{\Longrightarrow} F$ and $E \overset{u}{\Longrightarrow}$, where $u \in \Sigma^*$, are then the obvious generalizations: $E \overset{\varepsilon}{\Longrightarrow} F$ iff $E \overset{\tau}{\Longrightarrow} F$, and for $u = a_1 a_2 \cdots a_n \neq \varepsilon$, $E \overset{u}{\Longrightarrow} F$ iff there are states $E_1, E_2, ..., E_{n-1}$ such that $E \overset{a_1}{\Longrightarrow} E_1 \overset{a_2}{\Longrightarrow} \cdots \overset{a_{n-1}}{\Longrightarrow} E_{n-1} \overset{a_n}{\Longrightarrow} F$; and $E \overset{u}{\Longrightarrow}$ iff $E \overset{u}{\Longrightarrow} F$ for some F. Finally, we introduce one further bit of notation: given a set S of states, we let $\overset{u}{\Longrightarrow}(S) = \{F : E \overset{u}{\Longrightarrow} F \text{ for some } E \in S\}$.

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

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

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 in 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 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;
- $F: (P \times T) \cup (T \times P) \rightarrow \{0, 1\}$ defines the set of *arcs*; $\langle x, y \rangle$ is an arc iff $F(x, y) = 1$;
- $\ell : T \rightarrow \Sigma$ 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 $\mathbb{N}$). For technical convenience, we treat only ordinary nets; nevertheless all of our arguments can be easily modified to hold for these more general nets. We display nets graphically using circles for places and boxes for transitions; when labels of transitions are important, we write them inside the boxes.

A *marking* of a net is a mapping $M: P \rightarrow \mathbb{N}$ associating a number of *tokens* to each place. We denote the zero marking, that is, the marking that maps each place to 0, by $\mathbf{0}$. A transition t is *enabled* at a marking M, written $M[t\rangle$, iff $M(p) \geqslant F(p, t)$ 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 extend this firing rule to sequences of transitions, thus

[4] This is somewhat nonstandard in process theory; our relations $\overset{a}{\Longrightarrow}$ should be written as $\overset{\hat{a}}{\Longrightarrow}$ in order to fit into the process theory framework [19], but in our presentation we omit the extra decoration.

writing $M[t_1 t_2 \cdots t_n\rangle M'$ when $M[t_1\rangle M_1 [t_2\rangle \cdots M_{n-1} [t_n\rangle M'$ for some $M_1, M_2, ..., M_{n-1}$ (and $M[t_1 t_2 \cdots t_n\rangle$ when $M[t_1 t_2 \cdots t_n\rangle M'$ for some M').

We 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 $\ell(t) = a$. Notions like $M \longrightarrow M'$, $M \overset{u}{\Longrightarrow} M'$, $\mathscr{T}(M)$, $M_1 \sim M_2$ are then inherited from the respective notions given in the general setting. In particular, we have the notion of the reachable markings of a marking M of a net N; in this case we write either $\mathscr{R}_N(M)$ or $\mathscr{R}(M)$ if the underlying net N is clear from the context. Observe that the LTS derived from a net is an admissible system, so Proposition 2.2 is applicable.

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

Theorem 2.3 (Mayr [18]). *Given two markings M and M' of a Petri net N, it is decidable whether or not $M \rightarrow^* M'$, that is, whether or not $M' \in \mathscr{R}(M)$.*

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

We define the ordering $\leqslant$ pointwise on the set $\mathbb{N}_\omega^P$ of ω-markings of a net with place set P, thus writing $\hat{M} \leqslant \hat{M}'$ iff $\hat{M}(p) \leqslant \hat{M}'(p)$ for every $p \in P$; this is a *partial order* on ω-markings (it is reflexive, transitive, and antisymmetric). Moreover, it satisfies the *finite basis property* (*fbp*): every infinite sequence of elements has an infinite (not necessarily strictly) ascending subsequence. This result is known as Dickson's lemma [2].

Lemma 2.4 (Dickson [2]). *The collection of ω-markings of a net ordered by $\leqslant$ satisfies the fbp. Specifically, given an infinite sequence of ω-markings $\hat{M}_1, \hat{M}_2, \hat{M}_3, ...,$ there are indices $i_1 < i_2 < i_3 < \cdots$, such that $\hat{M}_{i_1} \leqslant \hat{M}_{i_2} \leqslant \hat{M}_{i_3} \leqslant \cdots$.*

Proof. By induction on the number of places: for each place p in turn, we choose an infinite subsequence $\hat{M}_{i_1}, \hat{M}_{i_2}, \hat{M}_{i_3}, ...,$ such that $\hat{M}_{i_1}(p) \leqslant \hat{M}_{i_2}(p) \leqslant \hat{M}_{i_3}(p) \leqslant \cdots$. ∎

Finally, we extend the ordering to *sets* of ω-markings by defining $\mathscr{M} \leqslant \mathscr{M}'$ iff for every $\hat{M} \in \mathscr{M}$ there exists $\hat{M}' \in \mathscr{M}'$ with $\hat{M} \leqslant \hat{M}'$; this relation is a *preorder* on sets of ω-markings (it is reflexive and transitive, but not antisymmetric). We may then observe the following.

Lemma 2.5. (a) *If $\hat{M}'[t\rangle \hat{M}'_t$ and $\hat{M} \geqslant \hat{M}'$, then $\hat{M}[t\rangle \hat{M}_t$ (and hence, $\hat{M}_t \geqslant \hat{M}'_t$).*

(b) *If $\hat{M}' \overset{a}{\Longrightarrow} \hat{M}'_a$ and $\hat{M} \geqslant \hat{M}'$, then $\hat{M} \overset{a}{\Longrightarrow} \hat{M}_a$ with $\hat{M}_a \geqslant \hat{M}'_a$.*

(c) *If $\hat{M}'[u\rangle$ and $\hat{M}(p) \geqslant \min(|u|, \hat{M}'(p))$ for all places p, then $\hat{M}[u\rangle$.*

(d) *If $\hat{M}' \overset{w}{\Longrightarrow}$ and $\hat{M} \geqslant \hat{M}'$, then $\hat{M} \overset{w}{\Longrightarrow}$.*

(e) *If $\mathscr{M} \leqslant \mathscr{M}'$, then $\overset{a}{\Longrightarrow}(\mathscr{M}) \leqslant \overset{a}{\Longrightarrow}(\mathscr{M}')$.*

Proof. Part (a) is easily proved directly; part (b) is proved from part (a) by induction on the number of unobservable τ transitions involved in the transition $\hat{M}' \stackrel{a}{\Longrightarrow} \hat{M}'_a$; part (c) is proved by induction on $|u|$; part (d) follows from part (c); and part (e) follows from part (b). ∎

Every increasing chain $\hat{M}_1 \leqslant \hat{M}_2 \leqslant \hat{M}_3 \leqslant \cdots$ of ω-markings has a unique least upper bound $\hat{M}$ defined by $\hat{M}(p) = \lim_{n \to \infty} \hat{M}_n(p)$ for each place p. For a set $\mathscr{M}$ of ω-markings we define its *completion* $\mathscr{C}(\mathscr{M})$ to be $\mathscr{M}$ enriched by such least upper bounds; and we use $\max(\mathscr{M})$ to refer to the subset of maximal elements of $\mathscr{M}$. Formally, these are defined as

$$\mathscr{C}(\mathscr{M}) = \{\hat{M} : \hat{M} \text{ is the least upper bound of a (possibly constant-valued) chain } \hat{M}_1 \leqslant \hat{M}_2 \leqslant \cdots \text{ in } \mathscr{M}\}.$$

$$\max(\mathscr{M}) = \{\hat{M} \in \mathscr{M} : \text{for all } \hat{M}' \in \mathscr{M}, \text{ either } \hat{M} = \hat{M}' \text{ or } \hat{M} \nleqslant \hat{M}'\}.$$

The first important observation we can make is that there can only be finitely many maximal elements in any set of ω-markings.

LEMMA 2.6. *For every set $\mathscr{M}$ of ω-markings*, $\max(\mathscr{M})$ *is finite.*

Proof. Immediate from Lemma 2.4. ∎

We can then make the following sequence of observations.

LEMMA 2.7. (a) $\mathscr{M} \subseteq \mathscr{C}(\mathscr{M})$.

(b) $\mathscr{C}(\mathscr{C}(\mathscr{M})) = \mathscr{C}(\mathscr{M})$.

(c) $\max(\mathscr{M}) \subseteq \mathscr{M}$.

(d) *If $\mathscr{M} \subseteq \mathscr{M}'$, then $\mathscr{M} \leqslant \mathscr{M}'$.*

(e) *If $\mathscr{M} \subseteq \mathscr{M}'$, then $\stackrel{a}{\Longrightarrow}(\mathscr{M}) \subseteq \stackrel{a}{\Longrightarrow}(\mathscr{M}')$.*

(f) *If* $\max(\mathscr{M}) \leqslant \max(\mathscr{M}') \leqslant \max(\mathscr{M})$, *then* $\max(\mathscr{M}) = \max(\mathscr{M}')$.

Proof. Immediate. ∎

LEMMA 2.8. *For every $\hat{M} \in \mathscr{C}(\mathscr{M})$ there is $\hat{M}' \in \max(\mathscr{C}(\mathscr{M}))$ such that $\hat{M} \leqslant \hat{M}'$.*

Proof. By induction on the number of places p with $\hat{M}(p) < \infty$. If there is no chain $\hat{M} \leqslant \hat{M}_1 \leqslant \hat{M}_2 \leqslant \cdots$ of *distinct* ω-markings in $\mathscr{C}(\mathscr{M})$, then the result readily follows. Otherwise let $\hat{M}_\infty \in \mathscr{C}(\mathscr{M})$ be the least upper bound of such a chain. Then $\hat{M} \leqslant \hat{M}_\infty$, and $\hat{M}_\infty$ must have fewer finitely marked places than $\hat{M}$, from which the result follows by induction. ∎

LEMMA 2.9. *If $\mathscr{M} \leqslant \mathscr{M}'$, then* $\max(\mathscr{C}(\mathscr{M})) \leqslant \max(\mathscr{C}(\mathscr{M}'))$.

Proof. Let $\hat{M} \in \max(\mathscr{C}(\mathscr{M}))$. Then by Lemma 2.7(c), $\hat{M} \in \mathscr{C}(\mathscr{M})$, so $\hat{M}$ is the least upper bound of a chain $\hat{M}_1 \leqslant \hat{M}_2 \leqslant \cdots$ in $\mathscr{M}$. From $\mathscr{M} \leqslant \mathscr{M}'$ we can find a sequence $\hat{M}'_1, \hat{M}'_2, \ldots$ in $\mathscr{M}'$ with $\hat{M}_i \leqslant \hat{M}'_i$ for each $i \geqslant 1$. By Lemmas 2.7(a) and 2.8, we can find $\hat{M}''_i \in \max(\mathscr{C}(\mathscr{M}'))$ such that $\hat{M}'_i \leqslant \hat{M}''_i$ for each $i \geqslant 1$. By Lemma 2.6, there must be a single $\hat{M}' \in \max(\mathscr{C}(\mathscr{M}'))$ such that $\hat{M}'_i \leqslant \hat{M}'$ for infinitely many $i \geqslant 1$, and hence $\hat{M}_i \leqslant \hat{M}'$ for all $i \geqslant 1$. Thus $\hat{M} \leqslant \hat{M}'$. ∎

LEMMA 2.10. $\mathscr{C}(\bigcup_{1 \leqslant i \leqslant n} \mathscr{M}_i) = \bigcup_{1 \leqslant i \leqslant n} \mathscr{C}(\mathscr{M}_i)$.

Proof. For inclusion in the forward direction, any $\hat{M} \in \mathscr{C}(\bigcup_{1 \leqslant i \leqslant n} \mathscr{M}_i)$ must be the least upper bound of a chain $\hat{M}_1 \leqslant \hat{M}_2 \leqslant \cdots$ of markings taken from $\bigcup_{1 \leqslant i \leqslant n} \mathscr{M}_i$; but since there are only finitely many $\mathscr{M}_i$, we must be able to take a subchain $\hat{M}_{i_1} \leqslant \hat{M}_{i_2} \leqslant \cdots$ of markings from a single $\mathscr{M}_i$, from which we deduce that $\hat{M} \in \mathscr{C}(\mathscr{M}_i) \subseteq \bigcup_{1 \leqslant i \leqslant n} \mathscr{C}(\mathscr{M}_i)$.

For inclusion in the reverse direction, any $\hat{M} \in \bigcup_{1 \leqslant i \leqslant n} \mathscr{C}(\mathscr{M}_i)$ must come from some $\mathscr{C}(\mathscr{M}_i)$, from which we get $\hat{M} \in \mathscr{C}(\bigcup_{1 \leqslant i \leqslant n} \mathscr{M}_i)$. ∎

LEMMA 2.11. $\max(\bigcup_i \mathscr{M}_i) = \max(\bigcup_i \max(\mathscr{M}_i))$.

Proof. $\hat{M} \in \max(\bigcup_i \mathscr{M}_i)$

iff $\hat{M} \in \mathscr{M}_i$ for some i, and $\hat{M} \nleqslant \hat{M}'$ for every $\hat{M}' \neq \hat{M}$ in $\bigcup_i \mathscr{M}_i$

iff $\hat{M} \in \max(\mathscr{M}_i)$ for some i, and $\hat{M} \nleqslant \hat{M}'$ for every $\hat{M}' \neq \hat{M}$ in $\bigcup_i \max(\mathscr{M}_i)$

iff $\hat{M} \in \max(\bigcup_i \max(\mathscr{M}_i))$. ∎

LEMMA 2.12. $\overset{a}{\Longrightarrow}(\mathscr{C}(\mathscr{M})) \subseteq \mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M}))$.

Proof. Let $\hat{M}_a \in \overset{a}{\Longrightarrow}(\mathscr{C}(\mathscr{M}))$. Then there is a chain $\hat{M}_1 \leqslant \hat{M}_2 \leqslant \cdots$ in $\mathscr{M}$ with least upper bound $\hat{M}$ such that $\hat{M} \xrightarrow{w} \hat{M}_a$, where $w = \tau^k a \tau^\ell$ $(k, \ell \geqslant 0)$ if $a \neq \tau$ and $w = \tau^k$ $(k \geqslant 0)$ if $a = \tau$. We can assume (by dropping a sufficient initial segment of the chain) that $\hat{M}_1(p) = \hat{M}(p)$ whenever $\hat{M}(p) < \infty$ and that $\hat{M}_1(p) \geqslant |w|$ whenever $\hat{M}(p) = \infty$. By Lemma 2.5(c) we have a sequence $\hat{M}'_1, \hat{M}'_2, \ldots$ in $\overset{a}{\Longrightarrow}(\mathscr{M})$ such that $\hat{M}_i \xrightarrow{w} \hat{M}'_i$ for each $i \geqslant 1$, all using the same sequence of net transistions as $\hat{M} \xrightarrow{w} \hat{M}_a$, thus having the same effect on the markings; in particular, $\hat{M}'_1 \leqslant \hat{M}'_2 \leqslant \cdots$. This chain has a least upper bound $\hat{M}'$ in $\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M}))$; it remains to demonstrate that $\hat{M}_a = \hat{M}'$, that is, that $\hat{M}_a(p) = \hat{M}'(p)$ for all places p:

- If $\hat{M}(p) < \infty$ then $\hat{M}(p) = \hat{M}_i(p)$ for all $i \geqslant 1$, so $\hat{M}_a(p) = \hat{M}'_i(p)$ for all $i \geqslant 1$, so $\hat{M}_a(p) = \lim_{i \to \infty} \hat{M}'_i(p) = \hat{M}'(p)$.
- If $\hat{M}(p) = \infty$ then for each $n \geqslant 0$ there is an $i \geqslant 1$ such that $\hat{M}_i(p) > n + |w|$, so $\hat{M}'_i(p) > n$, and hence, $\hat{M}'(p) = \lim_{i \to \infty} \hat{M}'_i(p) = \infty$. ∎

LEMMA 2.13. (a) $\mathscr{T}(\hat{M}') \subseteq \mathscr{T}(\hat{M})$ *for any* ω*-markings* $\hat{M}$ *and* $\hat{M}'$ *with* $\hat{M} \geqslant \hat{M}'$.

(b) *Given an increasing chain* $\hat{M}_1 \leqslant \hat{M}_2 \leqslant \hat{M}_3 \leqslant \cdots$ *of* ω*-markings with least upper bound* $\hat{M}$, *we have that* $\bigcup_{i \geqslant 1} \mathscr{T}(\hat{M}_i) = \mathscr{T}(\hat{M})$.

(c) $\bigcup_{\hat{M} \in \mathscr{M}} \mathscr{T}(\hat{M}) = \bigcup_{\hat{M} \in \max(\mathscr{C}(\mathscr{M}))} \mathscr{T}(\hat{M})$.

Proof. Part (a) follows directly from Lemma 2.5(d). For part (b), inclusion in the forward direction follows from part (a) since $\hat{M} \geqslant \hat{M}_i$ for each i. Inclusion in the reverse direction holds since whenever $\hat{M} \xrightarrow{w}$ we must have some $\hat{M}_i$ such that $\hat{M}_i(p) \geqslant \min(|w|, \hat{M}(p))$ for all places p; the result then follows from Lemma 2.5(c).

Finally for part (c), to show inclusion in the forward direction, let $\hat{M} \in \mathscr{M}$. By Lemmas 2.7(a) and 2.8 there is $\hat{M}' \in \max(\mathscr{C}(\mathscr{M}))$ with $\hat{M} \leqslant \hat{M}'$. Then by part (a), $\mathscr{T}(\hat{M}) \subseteq \mathscr{T}(\hat{M}')$. To show inclusion in the reverse direction let $\hat{M} \in \max(\mathscr{C}(\mathscr{M}))$. Thus by Lemma 2.7(c) $\hat{M} \in \mathscr{C}(\mathscr{M})$, so there is a chain $\hat{M}_1 \leqslant \hat{M}_2 \leqslant \cdots$ in $\mathscr{M}$ with least upper bound $\hat{M}$. By part (b), $\mathscr{T}(\hat{M}) = \bigcup_{i \geqslant 1} \mathscr{T}(\hat{M}_i) \subseteq \bigcup_{\hat{M} \in \mathscr{M}} \mathscr{T}(\hat{M})$. ∎

LEMMA 2.14. $\max(\mathscr{C}(\overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M}))))) = \max(\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M})))$.

Proof. By Lemma 2.7(f), it suffices to demonstrate

1. $\max(\mathscr{C}(\overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M}))))) \leqslant \max(\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M})))$;
2. $\max(\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M}))) \leqslant \max(\mathscr{C}(\overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M})))))$.

For 1 we have

$$\max(\mathscr{C}(\mathscr{M})) \subseteq \mathscr{C}(\mathscr{M}) \quad \text{(by Lemma 2.7(c))}$$

so

$$\begin{aligned} \overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M}))) &\subseteq \overset{a}{\Rightarrow}(\mathscr{C}(\mathscr{M})) && \text{(by Lemma 2.7(e))} \\ &\subseteq \mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M})) && \text{(by Lemma 2.12)} \end{aligned}$$

so

$$\overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M}))) \leqslant \mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M})) \qquad \text{(by Lemma 2.7(d))}$$

so

$$\begin{aligned} \max(\mathscr{C}(\overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M}))))) &\leqslant \max(\mathscr{C}(\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M})))) && \text{(by Lemma 2.9)} \\ &= \max(\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M}))) && \text{(by Lemma 2.7(b))} \end{aligned}$$

For 2 we have

$$\mathscr{M} \leqslant \max(\mathscr{C}(\mathscr{M})) \qquad \text{by Lemmas 2.7(a) and 2.8)}$$

so

$$\overset{a}{\Longrightarrow}(\mathscr{M}) \leqslant \overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M}))) \qquad \text{(by Lemma 2.5(e))}$$

so

$$\max(\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M}))) \leqslant \max(\mathscr{C}(\overset{a}{\Longrightarrow}(\max(\mathscr{C}(\mathscr{M}))))) \qquad \text{(by Lemma 2.9).} \quad \blacksquare$$

Given an ω-marking $\hat{M}$ of a net N, we can effectively find the (finitely many) maximal elements of $\mathscr{C}(\mathscr{R}(\hat{M}))$; this can be achieved by the technique of *coverability trees* [22]. Similarly, we can get the following.

LEMMA 2.15. *Given an ω-marking $\hat{M}$ and an action symbol $a \in \Sigma$, we can effectively construct* $\max(\mathscr{C}(\overset{a}{\Longrightarrow}\{\hat{M}\}))$. *Hence, we can effectively construct* $\max(\mathscr{C}(\overset{a}{\Longrightarrow}(\mathscr{M})))$ *for any finite set $\mathscr{M}$ of ω-markings.*

Proof. Assume that $a \neq \tau$. Construct net N' from N by removing all transitions with a label different from a or τ, and adding a new place p which is an input place to all transitions with label a. Then let $\hat{M}'$ have a single token on place p and let it be the same as $\hat{M}$ on the remaining places. Compute $\max(\mathscr{C}(\mathscr{R}_{N'}(\hat{M}')))$, take only the vectors in which the p-component is 0, and drop this 0-component from the vectors.

The case where $a = \tau$ is simpler. We construct N' by removing the non-τ transitions and compute $\max(\mathscr{C}(\mathscr{R}_{N'}(\hat{M}')))$. ∎

We shall also need an extension of Lemma 2.4 based on Higman's theorem [5].

THEOREM 2.16 (Higman [5]). *If a preorder* $\langle A, \leqslant \rangle$ *satisfies the fbp then so does* $\langle A^*, \underline{\ll} \rangle$, *where*

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

COROLLARY 2.17. *The collection* $\mathscr{P}_{\text{fin}}(\mathbb{N}_\omega^P)$ *of all finite sets of* ω*-markings for a net with place set* P *satisfies the fbp with respect to* $\leqslant$.

Proof. By Lemma 2.4, $\langle \mathbb{N}_\omega^P, \leqslant \rangle$ satisfies the fbp. Hence, $\langle (\mathbb{N}_\omega^P)^*, \underline{\ll} \rangle$ also satisfies the fbp. The corollary is then clear from the fact that any finite set $\mathscr{M}$ can be viewed as a string of its elements. ∎

Finally, we shall need the following additional technical result (for Theorem 4.8). Let us call a set of markings $\mathscr{M} \subseteq \mathbb{N}^P$ *simple* if there is a disjoint partition $P = P_1 \cup P_2$, a fixed mapping $fix : P_1 \to \mathbb{N}$, and a constant n such that

$$\mathscr{M} = \{M : M(p_1) = fix(p_1) \text{ and } M(p_2) \geqslant n \text{ for all } p_1 \in P_1 \text{ and } p_2 \in P_2\}.$$

The next result shows that it is semidecidable if a simple set $\mathscr{M}_2$ is reachable via markings from a simple set $\mathscr{M}_1$ whose nonfixed values can be arbitrarily large.

LEMMA 2.18. *The following problem is semidecidable*:

Instance: *A marking* M_0 *of a net* N, *and two simple sets* $\mathscr{M}_1$ *and* $\mathscr{M}_2$ *of markings of* N, *where* $P = P_1 \cup P_2$ *is the partition relevant to* $\mathscr{M}_1$.

Question: *For every* $m \in \mathbb{N}$, *does there exist* $M_m \in \mathscr{M}_1$ *with* $M_m(p_2) > m$ *for every* $p_2 \in P_2$ *such that* $M_0 \to^* M_m \to^* \mathscr{M}_2$?

Proof. Consider an instance of the problem as given above. We use u, v, w to denote transition sequences of N and Δ_u to denote the mapping (of places to integers) indicating the change in the marking upon performing u; that is, if $M\,[u\rangle\, M'$ then $M' = M + \Delta_u$.

We define a preorder $\langle \mathscr{D}, \preccurlyeq \rangle$, where

$$\mathscr{D} = \{\langle u, M_1, v, M_2 \rangle : M_1 \in \mathscr{M}_1, M_2 \in \mathscr{M}_2, \text{ and } M_0\,[u\rangle\, M_1\,[v\rangle\, M_2\}$$

and $\langle u, M_1, v, M_2 \rangle \preccurlyeq \langle u', M_1', v', M_2' \rangle$ iff we can write

$$\begin{array}{ll} u = u_1 \quad u_2 \quad u_3 \cdots u_k & v = v_1 \quad v_2 \quad v_3 \cdots v_\ell \\ u' = u_1 w_1 u_2 w_2 u_3 \cdots u_k w_k & v' = v_1 w_{k+1} v_2 w_{k+2} v_3 \cdots v_\ell w_{k+\ell}, \end{array}$$

so that $\Delta_w \geqslant \mathbf{0}$ for every prefix $w = w_1 w_2 \cdots w_i$ $(0 \leqslant i \leqslant k+\ell)$ of $w_1 w_2 \cdots w_i \cdots w_{k+\ell}$. By a variation of the proof of Theorem 6.5 in [9] (and using Theorem 2.16 as in that proof), we can show that $\langle \mathscr{D}, \preccurlyeq \rangle$ satisfies the fbp. (Briefly, the proof works by providing a straightforward order-preserving encoding of the elements of $\mathscr{D}$ as sequences of vectors over $\mathbb{N}$ and then recalling that vectors over $\mathbb{N}$ satisfy the fbp.)

Given $\langle u, M_1, v, M_2 \rangle \preccurlyeq \langle u', M'_1, v', M'_2 \rangle$ with the sequences as above for any prefix $s^{(j)}$ of $u_1 w_1^j u_2 w_2^j \cdots u_k w_k^j v_1 w_{k+1}^j v_2 w_{k+2}^j \cdots v_\ell w_k^\ell$, we get (from Lemma 2.5(c) applied repeatedly) that $M_0 [\, s^{(j)} \rangle\, M^{(j)}$ for some $M^{(j)}$ and that $M^{(j)} \leqslant M^{(k)}$ for $j \leqslant k$. Thus we can "pump" the w_i's; that is, for every $j \geqslant 0$,

$$M_0 \,[u_1 w_1^j u_2 w_2^j \cdots u_k w_k^j\rangle\, M_1^{(j)} \,[v_1 w_{k+1}^j v_2 w_{k+2}^j \cdots v_\ell w_{k+\ell}^j\rangle\, M_2^{(j)}.$$

Note that $M_1^{(j)} = M_1 + j \cdot \Delta_{w_1 w_2 \cdots w_k}$, so $\Delta_{w_1 w_2 \cdots w_k}(p_1) = 0$ for every $p_1 \in P_1$, so $M_1^{(j)} \in \mathscr{M}_1$ for every $j \in \mathbb{N}$; similarly, $M_2^{(j)} \in \mathscr{M}_2$ for every $j \in \mathbb{N}$.

We call a pair $\langle u, M_1, v, M_2 \rangle \preccurlyeq \langle u', M'_1, v', M'_2 \rangle$ *useful* if $M_1(p_2) < M'_1(p_2)$ for every $p_2 \in P_2$ (that is, $\Delta_{w_1 w_2 \cdots w_k}(p_2) > 0$ for every $p_2 \in P_2$). The existence of a useful pair is obviously a sufficient condition for a positive answer to the considered problem instance (since the relevant w_i's can be pumped indefinitely).

Now observe that if we have a positive answer to our problem, then there must be an infinite sequence

$$\langle u^{(1)}, M_1^{(1)}, v^{(1)}, M_2^{(1)} \rangle, \langle u^{(2)}, M_1^{(2)}, v^{(2)}, M_2^{(2)} \rangle, \langle u^{(3)}, M_1^{(3)}, v^{(3)}, M_2^{(3)} \rangle, \ldots$$

of elements of $\mathscr{D}$ such that for every n and every $p_2 \in P_2$ we have $M_1^{(n)}(p_2) < M_1^{(n+1)}(p_2)$. Thus, the existence of a useful pair is guaranteed due to the fbp, so it is also a necessary condition for a positive answer.

The desired semidecidability is then clear due to the possibility of generating all pairs $\langle u, M_1, v, M_2 \rangle, \langle u', M'_1, v', M'_2 \rangle \in \mathscr{D}$ and checking if any constitutes a useful pair. ∎

3. TRACE EQUIVALENCE

In this section we demonstrate the decidability of the trace equivalence problem and the undecidability of the trace finiteness problems.

3.1. *Decidability of (Strong and Weak) Trace Equivalence*

Here we demonstrate the decidability of the following:

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

To do this, we show decidability for the trace inclusion problem in both directions, $\mathscr{T}(M_0) \subseteq \mathscr{T}(r_0)$ and $\mathscr{T}(r_0) \subseteq \mathscr{T}(M_0)$. Without loss of generality we suppose that R has no τ labels and is deterministic; that is, 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 ε-move elimination and subset construction algorithms for nondeterministic finite automata (cf., e.g., [7]).

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

Proof. First, we can observe the semidecidability of the complementary problem $\mathcal{T}(M_0) \nsubseteq \mathcal{T}(r_0)$. For this, it suffices to generate all sequences from $(\Sigma \backslash \{\tau\})^*$ and to stop when some $w \in \mathcal{T}(M_0) \backslash \mathcal{T}(r_0)$ is found; semidecidability of this final test is obvious.

Now define the binary relation $S = \{\langle \hat{M}, r \rangle : \mathcal{T}(\hat{M}) \subseteq \mathcal{T}(r)\}$ between ω-markings of the net and states of the LTS and define the ordering $\leqslant$ on S by $\langle \hat{M}, r \rangle \leqslant \langle \hat{M}', r' \rangle$ iff $\hat{M} \leqslant \hat{M}'$ and $r = r'$. By Lemma 2.6, S has finitely many maximal elements, and by Lemmas 2.8 and 2.13(a), S is the downwards closure of these maximal elements. Now observe the following simple fact: If a set X of pairs $\langle \hat{M}, r \rangle$ satisfies the condition

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

then $X \subseteq S$.

It is also clear from Lemma 2.5(a) (along with Lemmas 2.7(a) and 2.8) that if X is the downwards closure of its maximal elements then it suffices to verify (∗) only for its maximal elements.

Since S satisfies (∗) (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 $\langle \hat{M}, r \rangle$ such that its downwards closure satisfies (∗) 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 $\mathcal{T}(M_0) \subseteq \mathcal{T}(r_0)$. ∎

THEOREM 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. Each node of the tree is labelled by a pair $\langle r, \mathcal{M} \rangle$, where r is a state of R and $\mathcal{M}$ is a set of pairwise incomparable ω-markings of N (and, hence, is finite). The edges in the tree are labelled by $\Sigma \backslash \{\tau\}$. The tree is defined inductively as

1. We start with the root node which we label $\langle r_0, \{M_0\} \rangle$.

2. From a given node $\langle r, \mathcal{M} \rangle$, we add one a-labelled edge for each transition $r \xrightarrow{a} r'$. (By our assumption on R, $a \neq \tau$ and r' is uniquely determined by a.) This edge leads to a node which we label by $\langle r', \mathcal{M}' \rangle = \langle r', \max(\mathcal{C}(\xRightarrow{a}(\mathcal{M}))) \rangle$. Note that by Lemma 2.15, we can construct $\max(\mathcal{C}(\xRightarrow{a}\{\hat{M}\}))$ for each $\hat{M} \in \mathcal{M}$ and then take the maximal elements amongst all of these. This gives us our desired $\mathcal{M}'$, since

$$\begin{aligned}\mathscr{M}' &= \max(\mathscr{C}(\stackrel{a}{\Rightarrow}(\mathscr{M})))\\ &= \max\left(\mathscr{C}\left(\bigcup_{\hat{M}\in\mathscr{M}} \stackrel{a}{\Rightarrow}\{\hat{M}\}\right)\right)\\ &= \max\left(\bigcup_{\hat{M}\in\mathscr{M}} \mathscr{C}(\stackrel{a}{\Rightarrow}\{\hat{M}\})\right) && \text{(by Lemma 2.10)}\\ &= \max\left(\bigcup_{\hat{M}\in\mathscr{M}} \max(\mathscr{C}(\stackrel{a}{\Rightarrow}\{\hat{M}\}))\right) && \text{(by Lemma 2.11).}\end{aligned}$$

3. A node $\langle r, \mathscr{M}\rangle$ will be deemed a *leaf* (that is, we do not add the edges described in step 2) if

— $\mathscr{M}=\emptyset$ (in which case the leaf is deemed to be *unsuccessful*); or

— $\mathscr{M}\neq\emptyset$, and either r has no successors in R, or there is an ancestor $\langle r, \mathscr{M}'\rangle$ with $\mathscr{M}'\leqslant\mathscr{M}$ (in which case the leaf is deemed to be *successful*).

By Corollary 2.17, this tree must be finite, and thus, 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 $\mathscr{T}(r_0)\not\subseteq\mathscr{T}(M_0)$; otherwise $\mathscr{T}(r_0)\subseteq\mathscr{T}(M_0)$. To verify the correctness of this algorithm, we first note the following.

Claim. For any node $\langle r, \mathscr{M}\rangle$ in the tree reached from the root $\langle r_0, \{M_0\}\rangle$ by a path labelled by $w\neq\varepsilon$, we have $r_0 \xrightarrow{w} r$ and $\mathscr{M}=\max(\mathscr{C}(\stackrel{w}{\Rightarrow}\{M_0\}))$.

Proof of the claim. It is clear from the construction that $r_0\xrightarrow{w} r$. We prove that $\mathscr{M}=\max(\mathscr{C}(\stackrel{w}{\Rightarrow}\{M_0\}))$ by induction on $|w|$. For $w=a$, the result follows from the definition of $\mathscr{M}$. Suppose then that $\langle r_0, \{M_0\}\rangle \xrightarrow{w} \langle r, \mathscr{M}\rangle \xrightarrow{a} \langle r', \mathscr{M}'\rangle$. Then,

$$\begin{aligned}\mathscr{M}' &= \max(\mathscr{C}(\stackrel{a}{\Rightarrow}(\mathscr{M})))\\ &= \max(\mathscr{C}(\stackrel{a}{\Rightarrow}(\max(\mathscr{C}(\stackrel{w}{\Rightarrow}\{M_0\}))))) && \text{(by induction)}\\ &= \max(\mathscr{C}(\stackrel{a}{\Rightarrow}(\stackrel{w}{\Rightarrow}\{M_0\}))) && \text{(by Lemma 2.14)}\\ &= \max(\mathscr{C}(\stackrel{wa}{\Longrightarrow}(\{M_0\}))). \quad \blacksquare\end{aligned}$$

Certainly if $\mathscr{M}\neq\emptyset$ then $\mathscr{M}\subseteq\mathscr{C}(\mathscr{M})\neq\emptyset$, so (by Lemma 2.8) $\max(\mathscr{C}(\mathscr{M}))\neq\emptyset$. Thus, if we have an unsuccessful leaf $\langle r, \max(\mathscr{C}(\stackrel{w}{\Rightarrow}\{M_0\}))\rangle=\langle r,\emptyset\rangle$ at the end of a path labelled w, then $\stackrel{w}{\Rightarrow}\{M_0\}=\emptyset$; that is, $w\notin\mathscr{T}(M_0)$, whereas $r\in\mathscr{T}(r_0)$, so indeed, $\mathscr{T}(r_0)\not\subseteq\mathscr{T}(M_0)$.

Suppose then that all leaves are successful, and, despite this, $\mathscr{T}(r_0)\not\subseteq\mathscr{T}(M_0)$. Choose some $w\in\mathscr{T}(r_0\backslash\mathscr{T}(M_0)$ of minimal length. It can be written as $w=u_1u_2v$ with $u_2\neq\varepsilon$, where the tree has the path $\langle r_0, \{M_0\}\rangle \xrightarrow{u_1} \langle r, \mathscr{M}'\rangle \xrightarrow{u_2} \langle r, \mathscr{M}\rangle$ with $\mathscr{M}'\leqslant\mathscr{M}$. Then we must have that $v\in\mathscr{T}(r)$, but $v\notin\bigcup_{\hat{M}\in\stackrel{u_1u_2}{\Longrightarrow}\{M_0\}}\mathscr{T}(\hat{M})=\bigcup_{\hat{M}\in\mathscr{M}}\mathscr{T}(\hat{M})$ (by Lemma 2.13(c)). But then, by Lemma 2.13(a), $v\notin\bigcup_{\hat{M}\in\mathscr{M}'}\mathscr{T}(\hat{M})$ $=\bigcup_{\hat{M}\in\stackrel{u_1}{\Rightarrow}\{M_0\}}\mathscr{T}(\hat{M})$ (again by Lemma 2.13(c)), so $u_1v\in\mathscr{T}(r_0)\backslash\mathscr{T}(M_0)$, which contradicts the minimality of the length of w. $\blacksquare$

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 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 [13]). 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 [23] 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

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 follows from 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,

$$\mathtt{X}_1 : \mathrm{comm}_1$$

$$\mathtt{X}_2 : \mathrm{comm}_2$$

$$\dots$$

$$\mathtt{X}_{n-1} : \mathrm{comm}_{n-1}$$

$$\mathtt{X}_n : \mathtt{halt}$$

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

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

or of the form

$$X : \mathtt{if}\ \mathtt{c}_j = 0\ \mathtt{then}\ \mathtt{goto}\ X'$$
$$\mathtt{else}\ \mathtt{c}_j := \mathtt{c}_j - 1;\ \mathtt{goto}\ X''.$$

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 $\mathtt{X}_k$ $(1 \leqslant k < n)$, 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 [20]; there is no algorithm which decides whether or not a given Minsky machine halts.

Given a Minsky machine C, we define the net $N_C = \langle P, T, F, \Sigma, \ell \rangle$ with initial marking M_0 as

- The set of places is $P = \{c_1, c_2, ..., c_m, X_1, X_2, ..., X_n, U\}$.

- The initial marking M_0 will consist of just one token, located on the place X_1; 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

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

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

- For every instruction of the form

$$X: \mathtt{if}\ \mathtt{c}_j = 0\ \mathtt{then}\ \mathtt{goto}\ X'$$
$$\mathtt{else}\ \mathtt{c}_j := \mathtt{c}_j - 1;\ \mathtt{goto}\ X''$$

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

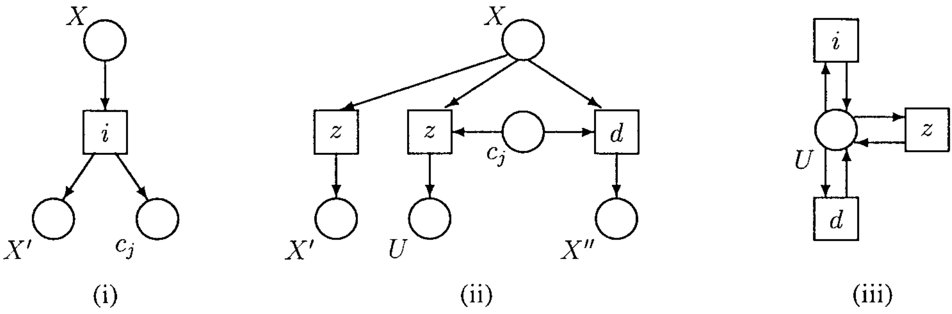

FIG 1. Constructions for N_C.

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 Fig.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 Fig. 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 well to the universal place from which any action is possible forevermore. Thus, $\mathscr{T}(M_0)$ consists of all "valid computation sequences" (that is, all prefixes of the computation of C) plus all sets $wz\{i, d, z\}^*$ such that wd is a valid computation sequence. From this it is clear that $\mathscr{T}(M_0)$ is regular if C halts; this fact is more carefully demonstrated in the following.

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 consists of a (finite) path corresponding to the (valid) computation of C (which halts by assumption); see Fig. 2. The states of R correspond to markings of N_C; and the initial state of this path is r_0 and corresponds to the initial marking M_0 of N_C. Outside of this path there is 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 have a further arc labelled by z leading to the state u.

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

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

$$X_i\colon \mathrm{comm}_i,$$

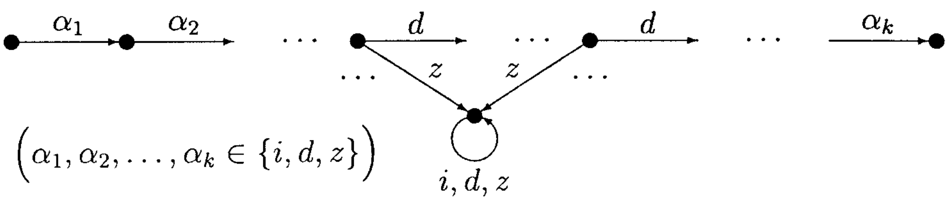

FIG. 2. Construction for R.

by the sequence of eight instructions:

$$
\begin{array}{lll}
X_i: & a_1 := a_1 + 1;\ \texttt{goto}\ Y_i^1 & * \ \textit{increment } a_1 \\
Y_i^1: & \texttt{if}\ a_1 = 0\ \texttt{then goto}\ Y_i^3 & * \ \text{while } a_1 > 0 \text{ do} \\
 & \texttt{else}\ a_1 := a_1 - 1;\ \texttt{goto}\ Y_i^2 & * \qquad \textit{decrement } a_1 \\
Y_i^2: & a_2 := a_2 + 1;\ \texttt{goto}\ Y_i^1 & * \qquad \textit{increment } a_2 \\
Y_i^3: & \texttt{if}\ a_2 = 0\ \texttt{then goto}\ Y_i^6 & * \ \text{while } a_2 > 0 \text{ do} \\
 & \texttt{else}\ a_2 := a_2 - 1;\ \texttt{goto}\ Y_i^4 & * \qquad \textit{decrement } a_2 \\
Y_i^4: & a_1 := a_1 + 1;\ \texttt{goto}\ Y_i^5 & * \qquad \textit{increment } a_1 \\
Y_i^5: & a_3 := a_3 + 1;\ \texttt{goto}\ Y_i^3 & * \qquad \textit{increment } a_3 \\
Y_i^6: & \texttt{if}\ a_3 = 0\ \texttt{then goto}\ Y_i^7 & * \ \text{while } a_3 > 0 \text{ do} \\
 & \texttt{else}\ a_3 := a_3 - 1;\ \texttt{goto}\ Y_i^6 & * \qquad \text{decrement } a_3 \\
Y_i^7: & \text{comm}_i &
\end{array}
$$

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 $\mathscr{T}(M_0)$ is different from the trace set of any finite-state process r_0.*

Proof. Suppose that $\mathscr{T}(M_0) = \mathscr{T}(r_0)$ 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 of C 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 of C from which it can follow an arbitrary number of decrement actions, which clearly is not possible for N_C starting in M_0. Hence, $\mathscr{T}(M_0) \neq \mathscr{T}(r_0)$ 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

In this section we demonstrate the decidability of the strong bisimulation equivalence and finiteness problems, and the undecidability of the weak bisimulation equivalence and finiteness problems. We start by describing a general decision technique which we shall use.

Given a transition system $\mathscr{L} = \langle \mathscr{S}, \Sigma, \{\xrightarrow{a}\}_{a \in \Sigma} \rangle$, we define the class of all *n-incompatible processes* (taken from other transition systems) as $\text{Inc}_n^{\mathscr{L}} = \{E : \forall F \in \mathscr{S} : E \not\sim_n F\}$. With this concept defined, the following useful observations can be made.

Proposition 4.1. *For any n, $E \sim F$ implies that $E \sim_n F$ and $E \not\rightarrow^* \text{Inc}_n^{\mathscr{L}(F)}$. In addition, the reverse implication holds under the further proviso that $\sim_{n-1}$ coincides with $\sim_n$ (and hence, with $\sim$) over $\mathscr{L}(F)$.*

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 $\mathscr{L}(F)$, the relation

$$\{\langle E', F' \rangle : E' \in \mathscr{L}(E), F' \in \mathscr{L}(F), E' \sim_n F', E' \not\rightarrow^* \text{Inc}_n^{\mathscr{L}(F)}\}$$

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

Corollary 4.2. *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,*

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

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

Corollary 4.3. *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 $E \not\rightarrow^* \text{Inc}_n^R$.*

Proof. Immediate. ▮

Further development and applications of this technique are presented in [15, 16].

Before we proceed with our decidability proofs, we define a few further useful concepts and make various important observations. We say that a marking L of a net N is *n-bounded* iff $L(p) \leqslant n$ for each place p. For every n-bounded marking L, we define $L^{\geqslant n}$ to be the set of all markings M such that $L(p) = \min(n, M(p))$ for each place p, and we note the following.

Lemma 4.4. *For every n-bounded marking L and every $M \in L^{\geqslant n}$, $L \sim_n M$.*

Proof. By a simple induction on n. ▮

Note that for every marking M there is a unique n-bounded marking L_M, defined by $L_M(p) = \min(n, M(p))$, such that $M \in L_M^{\geqslant n}$; that is, $M \in L^{\geqslant n}$ iff $L = L_M$. Also, there are clearly only finitely many n-bounded markings, all of which we may effectively list.

Next, given a net N and an n-state LTS R, we let

$$\langle N, R \rangle\text{-Inc} = \text{Inc}_n^R \cap \{M : M \text{ is a marking of } N\}.$$

This is the set of markings of N which are not strongly n-bisimilar to (that is, not in the relation $\sim_n$ with) any state of R. By Lemma 4.4, $M \sim_n L_M$, so $M \in \langle N, R\rangle$-Inc iff $L_M^{\geqslant n} \subseteq \langle N, R\rangle$-Inc. Hence, $\langle N, R\rangle$-Inc can be expressed as the union

$$\langle N, R\rangle\text{-Inc} = L_1^{\geqslant n} \cup L_2^{\geqslant n} \cup \cdots \cup L_k^{\geqslant n},$$

where $L_1, L_2, ..., L_k$ are all of the n-bounded markings appearing in $\langle N, R\rangle$-Inc; we can effectively construct this union, since by Proposition 2.2 we can decide if each n-bounded marking L is in $\langle N, R\rangle$-Inc.

4.1. Decidability of Strong Bisimulation Equivalence

The decidability proof for strong bisimulation equivalence is based on the general method described above. Given a marking M_0 of a net N and a state r_0 of an n-state LTS R, the question $M_0 \sim_n r_0$ is decidable (by Proposition 2.2). Therefore by Corollary 4.3, it suffices to show the decidability of the question as to whether the set $\langle N, R\rangle$-Inc is reachable from M_0. From the above characterisation of $\langle N, R\rangle$-Inc, it then suffices to show the decidability as to whether the set $L^{\geqslant n}$ is reachable from M_0, where L is an arbitrary n-bounded marking.

Theorem 4.5. *The problem $M_0 \sim r_0$ is decidable.*

Proof. From the above considerations, it suffices to show the decidability of the following:

Given an n-bounded marking L, is there some $M \in L^{\geqslant n}$ such that $M_0 \rightarrow^* M$?

But this problem is easily reducible to the reachability problem (Theorem 2.3); 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 net N is strongly bisimilar to some (unspecified) finite-state process. We refer to this problem as the *strong bisimulation finiteness problem*, or the *strong b-finiteness problem* for short.

Formally, we say that a marking M is *b-finite* iff $\mathscr{R}(M)$ contains only finitely many equivalence classes with respect to $\sim$; otherwise we say that M is *b-infinite*, that is, if there exist infinitely many markings $M_1, M_2, M_3, ...$ 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.

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 $P_2 \neq \varnothing$. For mappings $M_1 : P_1 \rightarrow \mathbb{N}$ and $M_2 : P_2 \rightarrow \mathbb{N}$, (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, $P_2 \neq \varnothing$ of P and mappings $M_1 : P_1 \rightarrow \mathbb{N}$, $M_2 : P_2 \rightarrow \mathbb{N}$." 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 $(M : P_1 \to \mathbb{N})$ of the *subnet* of N obtained by removing all places from P_2, together with their adjacent arcs; this is behaviourally equivalent to putting ω on all places of P_2. By Lemma 4.4 then, for any $n \geqslant 0$, if $M'(p) \geqslant n$ for each place p then $(M, M') \sim_n (M, -)$.

LEMMA 4.6. *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 $n \geqslant 0$ there is an index i such that $M_i(p) \geqslant n$ for each p. Then $(M, -) \sim_n (M, M_i)$ holds and, since $(M, M_1) \sim (M, M_i)$, we also have $(M, -) \sim_n (M, M_1)$. Therefore, $(M, -) \sim_n (M, M_1)$ for every $n \geqslant 0$, and so, by Proposition 2.1, $(M, -) \sim (M, M_1)$. ∎

LEMMA 4.7. *A marking* M_0 *is b-infinite iff there exists a marking* $(M, -)$ *and an increasing chain of markings* $M_1 < M_2 < M_3 < \cdots$ *with* $(M, M_i) \in \mathscr{R}(M_0)$ *for every* $i \geqslant 1$ *such that*

1. $(M, -)$ *is b-infinite; or*
2. $(M, -)$ *is b-finite and* $(M, M_i) \not\sim (M, -)$ *for every* $i \geqslant 1$.

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 2.4, there is an infinite subsequence (M, M_1), (M, M_2), (M, M_3), ... with $M_1 < M_2 < M_3 < \cdots$. If $(M, -)$ is b-infinite then case 1 holds. If $(M, -)$ is b-finite then we can assume that $(M, M_i) \not\sim (M, -)$ for every $i \geqslant 1$ (since the markings (M, M_i) are pairwise non-bisimilar, at most one of them can be bisimilar to $(M, -)$, and we can simply drop this marking from the sequence) and thus case 2 holds.

$(\Leftarrow)$ Let $\mathscr{M} = \{(M, M_i) : i \geqslant 1\}$. If $\mathscr{M}$ contains infinitely many pairwise non-bisimilar markings, then M_0 is b-infinite, and we are done. So assume that $\mathscr{M}$ contains infinitely many pairwise bisimilar markings. By Lemma 4.6, all of these markings must be bisimilar to $(M, -)$, and so case 2 cannot hold. Thus case 1 holds; that is, $(M, -)$ is b-infinite and, therefore, M_0 must be b-infinite, since it has a reachable marking which is bisimilar to $(M, -)$. ∎

THEOREM 4.8. *It is decidable whether or not a marking* M_0 *of a net* N *is strongly b-finite.*

Proof. By induction on the number of places of N. If N has no places, then M_0 is the unique mapping $M : \varnothing \to \mathbb{N}$, and it is certainly b-finite, since $\mathscr{R}(M_0) = \{M_0\}$. Assume then that N has some places. As noted above, it suffices to show semidecidability of the b-infiniteness problem; and for this, Lemma 4.7 shows that it suffices to enumerate all markings $(M, -)$ of N (for all partitions P_1, P_2 with $P_2 \neq \varnothing$), and to show that it is semidecidable whether or not there exists a chain as specified in the lemma such that one of the two conditions of the lemma holds.

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 $i \geqslant 1$ is surely semidecidable; just put $\mathcal{M}_1 = \mathcal{M}_2 = \{(M, M') : M' \text{ is arbitrary}\}$ and apply Lemma 2.18.

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 $i \geqslant 1$ is also semidecidable: if $(M, -)$ is b-finite then $(M, -) \sim r$ for a state r of some finite-state LTS R; we may assume that this R is known. (We may simply enumerate every finite state LTS R and decide whether $M \sim r$ for each state r of R.) Let n denote the number of states of R.

Claim. *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 $i \geqslant 1$ iff there exists an n-bounded marking L of N satisfying two conditions*:

(a) $L \in \langle N, R \rangle$-Inc; *and*

(b) *there exists a chain $M_1 < M_2 < M_3 < \cdots$ and markings $M'_1, M'_2, M'_3, \ldots \in L^{\geqslant n}$ such that $M_0 \rightarrow^* (M, M_i) \rightarrow^* M'_i$ for every $i \geqslant 1$.*

Proof of the claim. $(\Rightarrow)$ Let $M_1 < M_2 < M_3 < \cdots$ be a chain such that $(M, M_i) \in \mathcal{R}(M_0)$ and $(M, M_i) \not\sim (M, -)$ for every $i \geqslant 1$. There exists an index i_0 such that for every $i \geqslant i_0$, $M_i(p) \geqslant n$ for each p. For $i \geqslant i_0$ 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 $(M, M_i) \sim_n r$). Thus by Corollary 4.2, there exists an n-bounded marking $L_i \in \langle N, R \rangle$-Inc such that $(M, M_i) \rightarrow^* L_i^{\geqslant n}$. Since there are only finitely many n-bounded markings, there exists an n-bounded marking L and infinitely many indices $i_1 < i_2 < i_3 < \cdots$ such that $L = L_{i_1} = L_{i_2} = L_{i_3} = \cdots$. Thus L satisfies condition (a), and the subchain $M_{i_1} < M_{i_2} < M_{i_3} < \cdots$ satisfies condition (b).

$(\Leftarrow)$ Let M_i be an arbitrary marking of the chain given by (b); we need to prove that $(M, M_i) \not\sim (M, -)$. From (b) we have that $(M, M_i) \rightarrow^* L^{\geqslant n}$ and from (a) we have that $L^{\geqslant n} \subseteq \langle N, R \rangle$-Inc, so $(M, M_i) \rightarrow^* \langle N, R \rangle$-Inc. Hence by Corollary 4.2, $(M, M_i) \not\sim r$, so $(M, M_i) \not\sim (M, -)$. ∎

It remains to prove the semidecidability of conditions (a) and (b) for a given n-bounded marking L. Condition (a) is semidecidable by Proposition 2.2. For condition (b), set $\mathcal{M}_1 = \{(M, M') : M' \text{ is arbitrary}\}$ and $\mathcal{M}_2 = L^{\geqslant n}$ and apply Lemma 2.18. ∎

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 $\mathscr{R}_{N_1}(M) \subseteq \mathscr{R}_{N_2}(M)$?

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 assume that

- $|\mathscr{R}_{N_2}(M)| \geqslant 2$; and
- $\mathbf{0} \notin \mathscr{R}_{N_1}(M) \cap \mathscr{R}_{N_2}(M)$.

We shall describe the construction of a new net N with initial marking M_0 such that

1. if $\mathscr{R}_{N_1}(M) \nsubseteq \mathscr{R}_{N_2}(M)$ then $M_0 \approx r_1$, where r_1 is taken from the finite transition system R shown in Fig. 3; and
2. if $\mathscr{R}_{N_1}(M) \subseteq \mathscr{R}_{N_2}(M)$ 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.) We can note the following about the states of R:

- $r_4 \not\approx_1 r_9$, since $r_4 \overset{c}{\Longrightarrow}$ but $r_9 \overset{c}{\not\Longrightarrow}$.
- $r_3 \not\approx_2 r_7$, since $r_7 \overset{b}{\Longrightarrow} r_9$ would have to be matched by $r_3 \overset{b}{\Longrightarrow} r_4$, but $r_4 \not\approx_1 r_9$.
- $r_2 \not\approx_3 r_5$, since $r_5 \overset{a}{\Longrightarrow} r_7$ would have to be matched by $r_2 \overset{a}{\Longrightarrow} r_3$, but $r_3 \not\approx_2 r_7$.
- $r_1 \not\approx_4 r_5$, since $r_1 \overset{\tau}{\Longrightarrow} r_2$ would have to be matched by $r_5 \overset{\tau}{\Longrightarrow} r_5$, but $r_2 \not\approx_3 r_5$.

In particular, $r_1 \not\approx r_5$.

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

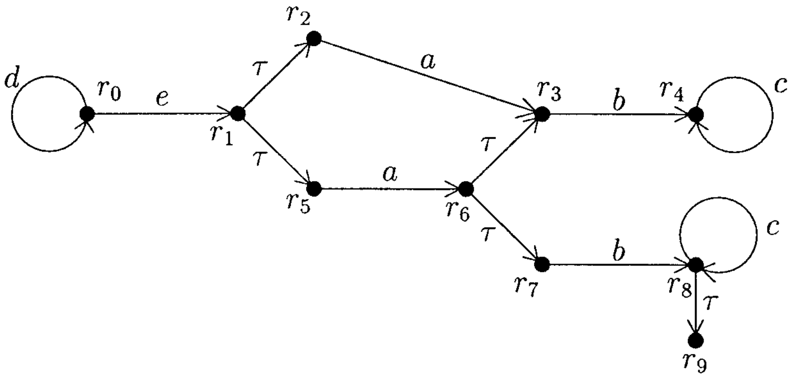

FIG. 3. The finite-state system R.

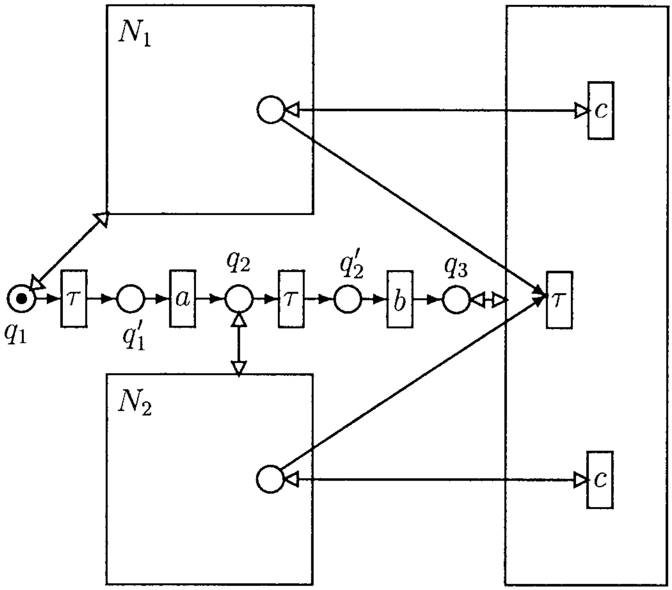

FIG. 4. Constructing the net N from N_1 and N_2.

N_i (for $i=1, 2$) are given by $P_i=\{p_i : p \in P\}$, and the transitions of N_i are given by $T_i=\{t_i : t \in T\}$. 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 4.9. 1. *If* $\mathcal{R}_{N_1}(M) \not\subseteq \mathcal{R}_{N_2}(M)$ *then* $M_0 \approx r_1$.

2. *If* $\mathcal{R}_{N_1}(M) \subseteq \mathcal{R}_{N_2}(M)$ *then* $M_0 \approx r_5$.

Proof. A weak bisimulation containing the pair $\langle M_0, r_1\rangle$ if $\mathcal{R}_{N_1}(M) \not\subseteq \mathcal{R}_{N_2}(M)$ and the pair $\langle M_0, r_5\rangle$ if $\mathcal{R}_{N_1}(M) \subseteq \mathcal{R}_{N_2}(M)$ consists of the following pairs. (In the following, we restrict M_1 to range only over $\mathcal{R}_{N_1}(M)$ and M_2 to range only over $\mathcal{R}_{N_2}(M)$; M_1' and M_2' are not so restricted.)

(a) $\langle (M_1, q_1, M), r_1\rangle$, where $\mathcal{R}_{N_1}(M_1) \not\subseteq \mathcal{R}_{N_2}(M)$ and $\mathcal{R}_{N_1}(M_1) \cap \mathcal{R}_{N_2}(M) \neq \varnothing$;

(b) $\langle (M_1, q_1, M), r_2\rangle$, where $\mathcal{R}_{N_1}(M_1) \cap \mathcal{R}_{N_2}(M) = \varnothing$;

(c) $\langle (M_1, q_1, M), r_5\rangle$, where $\mathcal{R}_{N_1}(M_1) \subseteq \mathcal{R}_{N_2}(M)$;

(d) $\langle (M_1, q_1', M), r_2\rangle$, where $M_1 \notin \mathcal{R}_{N_2}(M)$;

(e) $\langle (M_1, q_1', M), r_5\rangle$, where $M_1 \in \mathcal{R}_{N_2}(M)$;

(f) $\langle (M_1, q_2, M_2), r_3\rangle$, where $M_1 \notin \mathcal{R}_{N_2}(M_2)$;

(g) $\langle (M_1, q_2, M_2), r_6\rangle$, where $M_1 \in \mathcal{R}_{N_2}(M_2) \neq \{M_1\}$;

(h) $\langle (M_1, q_2, M_2), r_7\rangle$, where $\mathcal{R}_{N_2}(M_2) = \{M_1\}$;

(i) $\langle (M_1, q_2', M_2), r_3\rangle$, where $M_1 \neq M_2$;

(j) $\langle (M_1, q_2', M_2), r_7\rangle$, where $M_1 = M_2$;

(k) $\langle (M_1', q_3, M_2'), r_4\rangle$, where $M_1' \neq M_2'$;

(l) $\langle (M_1', q_3, M_2'), r_8\rangle$, where $M_1' = M_2' \neq \mathbf{0}$;

(m) $\langle (\mathbf{0}, q_3, \mathbf{0}), r_9\rangle$.

This collection of pairs does indeed constitute a weak bisimulation. We consider here only two interesting cases of matching transitions:

- If we take a pair from group (c): $\langle (M_1, q_1, M), r_5\rangle$, where $\mathcal{R}_{N_1}(M_1) \subseteq \mathcal{R}_{N_2}(M)$, then the transition $r_5 \xrightarrow{a} r_6$ is matched by the transition $(M_1, q_1, M) \overset{a}{\Longrightarrow} (M_1', q_2, M)$, where $M_1 \Rightarrow M_1'$; that is, $M_1' \in \mathcal{R}_{N_1}(M_1)$, so $M_1' \in \mathcal{R}_{N_1}(M_0)$ (since $M_1 \in \mathcal{R}_{N_1}(M_0)$), and $M_1' \in \mathcal{R}_{N_2}(M)$ (since $\mathcal{R}_{N_1}(M_1) \subseteq \mathcal{R}_{N_2}(M)$) and $\mathcal{R}_{N_2}(M) \neq \{M_1'\}$ (since by our earlier assumption, $|\mathcal{R}_{N_2}(M)| \geqslant 2$). Thus, the resulting pair is in group (g).
- If we take a pair from group (j): $\langle (M_1, q_2', M_2), r_7\rangle$, where $M_1 = M_2$, then the transition $r_7 \xrightarrow{b} r_8$ is matched by the transition $(M_1, q_2', M_2) \xrightarrow{b} (M_1, q_3, M_2)$. By our earlier assumption that $\mathbf{0} \notin \mathcal{R}_{N_1}(M) \cap \mathcal{R}_{N_2}(M)$, we have that $M_1 = M_2 \neq \mathbf{0}$. Thus, the resulting pair is in group (l).

The remaining cases are more readily verified. ∎

THEOREM 4.10. *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 4.9 and the fact established above that $r_1 \not\approx r_5$. ∎

Thus the problem $M \approx r_5$ is undecidable. Moreover, we may observe in the above proof that, since $r_1 \not\approx_4 r_5$, even the problem $M \approx_4 r_5$ 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; the state r_{fix} is then r_5.

4.4. Undecidability of Weak Bisimulation Finiteness

In this section we demonstrate the undecidability of the weak b-finiteness problem; that is, given a marking M_0 of a net 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 this problem is undecidable even when restricted to two counters both initialised with the value zero.

As already mentioned, we rely on a reduction from [10]. For our aims here, it suffices to recall that there is an algorithm given in the proof of Theorem 3.7 of [10, p. 291] which is specified as follows. (To understand this algorithm requires Definitions 3.2 and 3.5 of [10].)

Input: a 2-counter machine C.

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, satisfying $M(p^x) = M(p^y) = 0$. (In fact, N_1 and 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 $x, y \geqslant 0$,

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

Now let C be an arbitrary 2-counter machine. We construct another 2-counter machine C' which on input $(x, 0)$ runs as follows: first, it checks if $x = 2^k$ for some $k \geqslant 0$; 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 $(x, 0)$, $x \geqslant 0$;
- 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 Fig. 4; thus, we get a net N with a predefined initial marking M_0. We modify this net in the following way (depicted in Fig. 5). First, we remove the token from q_1. Second, we add the following new places and transitions:

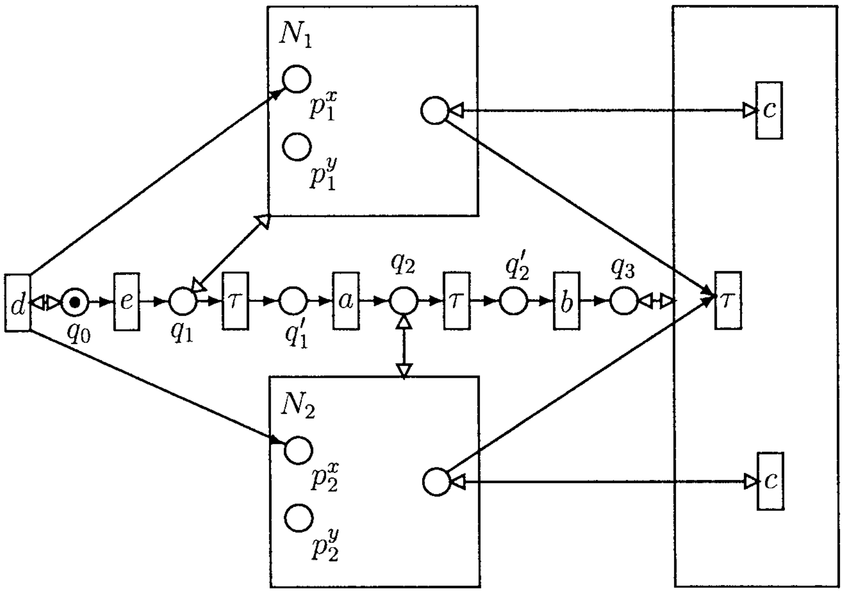

FIG. 5. Constructing the net N' from N.

- 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 (in particular, 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. It may then execute the e-transition, after which 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 4.11. *M_0' is weakly b-finite iff the* 2*-counter machine C halts on input* $(0, 0)$.

Proof. Let R be the finite-state transition system of Fig. 3.

($\Rightarrow$) Assume that C does not halt on input $(0, 0)$; thus C' halts on input $(x, 0)$ iff x is not a power of 2. Therefore, $\mathscr{R}_{N_1}(M^{x,0}) \subseteq \mathscr{R}_{N_2}(M^{x,0})$ iff $x = 2^k$ for some $k \geqslant 0$.

For any x, given the unique marking M reached after executing the d-transition x times in N', by Proposition 4.9 we have

1. if x is not a power of 2, then $M \overset{e}{\Longrightarrow} M'$ for some $M' \approx r_1$. (In fact, $M \overset{e}{\longrightarrow} M'$.)

2. if x is a power of 2, then there is no such M'. (If $M \overset{e}{\Longrightarrow} M'$ then $M' \approx r_5 \not\approx r_1$.)

We prove by contradiction that M'_0 is not weakly b-finite. 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 $r'_0 \xrightarrow{u} r'$, 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 $i \geqslant 0$, where $u = vwx$ and the projection of w onto the set of observable actions is a non-empty sequence of d's. Our contradiction then becomes apparent: by 1 we have that there is r'' in R' such that $r' \stackrel{e}{\Longrightarrow} r''$ and $r'' \approx r_1$; yet by 2 we have that there can be no such r''.

($\Leftarrow$) Assume that C halts on input $(0, 0)$; then C' halts on every input $(x, 0)$, $x \geqslant 0$. Therefore, after the occurrence of the e-transition we always have $\mathscr{R}_{N_1}(M^{x,0}) \nsubseteq \mathscr{R}_{N_2}(M^{x,0})$, regardless of the value of x. Hence, by the first clause of Proposition 4.9 it is clear that $M'_0 \approx r_0$, so M'_0 is weakly b-finite. ∎

THEOREM 4.12. *Neither the weak b-finiteness problem nor the weak b-infiniteness problem is semidecidable.*

Proof. By Proposition 4.11, 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. ∎

ACKNOWLEDGEMENT

We thank the anonymous referee for providing many helpful proposals for improving the presentation of various arguments in this paper.

REFERENCES

1. S. Christensen, Y. Hirshfeld, and F. Moller, Bisimulation is decidable for basic parallel processes, *in* "Proceedings of CONCUR'93: Concurrency Theory" (E. Best, Ed.), Lecture Notes in Computer Science, Vol. 715, pp. 143–157, Springer-Verlag, Berlin/New York, 1993.
2. L. E. Dickson, Finiteness of the odd perfect and primitive abundant numbers with distinct factors, *Amer. J. Math.* **35** (1913), 413–422.
3. R. J. van Glabbeek, The linear time-branching time spectrum, *in* "Proceedings of CONCUR'90: Concurrency Theory" (J. C. M. Baeten and J. W. Klop, Eds.), Lecture Notes in Computer Science, Vol. 458, pp. 278–297, Springer-Verlag, Berlin/New York, 1990.
4. M. Hack, The equality problem for vector addition systems is undecidable, *Theor. Comput. Sci.* **2** (1976), 77–95.
5. H. Higman, Ordering by divisibility in abstract algebras, *Proc. London Math. Soc.* **3**, No. 2 (1952), 326–336.
6. Y. Hirshfeld, Petri nets and the equivalence problem, *in* "Computer Science Logic: CSL'93" (E. Börger, Y. Gurevich, and K. Meinke, Eds.), Lecture Notes in Computer Science, Vol. 832, pp. 165–174, Springer-Verlag, Berlin/New York, 1994.
7. J. Hopcroft and J. D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison–Wesley, Reading, MA, 1979.

8. H. Hüttel, Undecidable equivalences for basic parallel processes, *in* "Proceedings of TACS'94: Theoretical Aspects of Computer Software" (M. Hagiya and J. C. Mitchell, Eds.), Lecture Notes in Computer Science, Vol. 789, pp. 454–464, Springer-Verlag, Berlin/New York, 1994.
9. P. Jančar, Decidability of a temporal logic problem for Petri nets, *Theor. Comput. Sci.* **74** (1990), 71–93.
10. P. Jančar, Undecidability of bisimilarity for Petri nets and some related problems, *Theor. Comput. Sci.* **148** (1995), 281–301.
11. P. Jančar, All action based behavioural equivalences are undecidable for labelled Petri nets, *Bull EATCS* **56** (1995), 86–88.
12. P. Jančar, Nonprimitive recursive complexity and undecidability for Petri net equivalences, *Theor. Comput. Sci.*, to appear.
13. P. Jančar, "Decidability Questions for Equivalences on Petri Nets," Habilitation thesis, Masaryk University, Brno, Czech Republic, 1995.
14. P. Jančar and J. Esparza, Deciding finiteness of Petri nets up to bisimulation, *in* "Proceedings of ICALP'96: Automata, Languages and Programming" (F. Meyer auf der Heide and B. Monien, Eds.), Lecture Notes in Computer Science, Vol. 1099, pp. 478–489, Springer-Verlag, Berlin/New York, 1996.
15. P. Jančar and A. Kučera, Bisimilarity of processes with finite-state systems, *in* "Proceedings of Infinity'97: the 2nd International Workshop on Verification of Infinite State Systems" (F. Moller, Ed.), [*Electronic Notes in Theor. Comput. Sci.* **9** (1997), URL: http://www.elsevier.nl/locate/entcs/volume9.html]
16. P. Jančar, A. Kučera, and R. Mayr, Deciding bisimulation-like equivalences with finite-state processes, *in* "Proceedings of ICALP'98: Automata, Languages and Programming" (K. G. Larsen, S. Skyum, and G. Winskel, Eds.), Lecture Notes in Computer Science, Vol. 1443, pp. 200–211, Springer-Verlag, Berlin/New York, 1998.
17. P. Jančar and F. Moller, Checking regular properties of Petri nets, *in* "Proceedings of CONCUR'95: Concurrency Theory" (I. Lee and S. A. Smolka, Eds.), Lecture Notes in Computer Science, Vol. 962, pp. 348–362, Springer-Verlag, Berlin/New York, 1995.
18. E. Mayr, An algorithm for the general Petri net reachability problem, *SIAM J. Comput.* **13** (1984), 441–460.
19. R. Milner, "Communication and Concurrency," Prentice–Hall, Englewood Cliffs, NJ, 1989.
20. M. Minsky, "Computation: Finite and Infinite Machines," Prentice–Hall, 1967.
21. E.-R. Olderog, "Nets, Terms, and Formulas," Cambridge Univ. Press, London, 1991.
22. J. L. Peterson, "Petri Net Theory and the Modeling of Systems," Prentice–Hall, 1981.
23. R. Valk and G. Vidal-Naquet, Petri nets and regular languages, *J. Comput. System Sci.* **23**, No. 3 (1981), 299–325.