Undecidability of Weak Bisimulation Equivalence for 1-Counter Processes

Richard Mayr

Department of Computer Science, Albert-Ludwigs-University Freiburg
Georges-Koehler-Allee 51, D-79110 Freiburg, Germany.
E-mail: mayrri@informatik.uni-freiburg.de
Fax: +49 761 203 8182

Abstract. We show that checking weak bisimulation equivalence of 1-counter nets (Petri nets with only one unbounded place) is undecidable. This implies the undecidability of weak bisimulation equivalence for 1-counter machines. The undecidability result carries over to normed 1-counter nets/machines.

Keywords: 1-counter nets, 1-counter machines, bisimulation

1 Introduction

Bisimulation equivalence plays a central role in the theory of process algebras [5]. The decidability and complexity of bisimulation problems for infinite-state systems has been studied intensively (see [10, 1] for surveys). Here we consider process models with a finite control and one unbounded counter (i.e., a register holding a natural number). There are 1-counter machines (Minsky counter machines [6] with only one counter) and 1-counter nets (Petri nets [7] with only one unbounded place). 1-counter nets are equivalent to the subclass of 1-counter machines where the counter cannot be tested for zero. 1-counter machines are equivalent to a subclass of pushdown automata (with only one stack symbol plus a bottom symbol that can never be removed).

The state of the art: Strong bisimilarity was shown to be decidable for 1-counter machines (and thus also for 1-counter nets) by Jančar [3] and later for general pushdown automata by Sénizergues [9]. Weak (and even strong) bisimilarity are undecidable for general Petri nets [2], but the proof in [2] uses several unbounded places. Weak bisimilarity was shown to be undecidable for pushdown automata by Srba [12]. The decidability of weak bisimilarity for the less expressive models of 1-counter machines and 1-counter nets was still open.

Our contribution. We show that weak bisimilarity is undecidable for 1-counter nets (even if they are normed). The undecidability of weak bisimilarity for 1-counter machines follows directly. This more general undecidability result subsumes the previously known undecidability of weak bisimilarity for PDA [12] and Petri nets [2].

2 Definitions

1-counter nets are Petri nets with only one unbounded place. We describe them in a simplified notation.

Definition 1. *A 1-counter net is given by (S, X, Act, Δ), where S is a finite set of control-states, X is a special symbol for the one unbounded place, Act is a finite set of atomic actions and Δ is a finite set of rewrite rules. The markings of this net are described in the form sX^n with $s \in S$ and $n \in \mathbb{N}_0$ (i.e., there are n tokens on the unbounded place and the configuration of the bounded rest of the net is s). The transitions of the net are described by the finite set Δ of rewrite rules of the form $s_1X^{m_1} \xrightarrow{a} s_2X^{m_2}$ with $s_1, s_2 \in S$, $m_1, m_2 \in \mathbb{N}_0$ and $a \in Act$. The labeled transition relation $\xrightarrow{a}$ on configurations (markings) of the 1-counter net is defined as follows: We have*

$$s_1X^n \xrightarrow{a} s_2X^{n-m_1+m_2}$$

iff there exists some rule $(s_1X^{m_1} \xrightarrow{a} s_2X^{m_2}) \in \Delta$ s.t. $n \geq m_1$. 1-counter nets are equivalent to 1-counter machines where the counter cannot be tested for 0.

We consider the semantic equivalence *weak bisimulation equivalence* (also called *weak bisimilarity*) [5] over labeled transition systems (e.g., those generated by 1-counter nets).

Definition 2. *The action τ is a special 'silent' internal action. The* extended transition relation '$\stackrel{a}{\Rightarrow}$' *is defined by $E \stackrel{a}{\Rightarrow} F$ iff either $E = F$ and $a = \tau$, or $E \xrightarrow{\tau^i} E' \xrightarrow{a} E'' \xrightarrow{\tau^j} F$ for some $i, j \in \mathbb{N}_0$. A binary relation R over states in a labeled transition graph is a* weak bisimulation *iff whenever $(E, F) \in R$ then for every $a \in Act$: if $E \xrightarrow{a} E'$ then there is $F \stackrel{a}{\Rightarrow} F'$ s.t. $(E', F') \in R$ and if $F \xrightarrow{a} F'$ then there is $E \stackrel{a}{\Rightarrow} E'$ s.t. $(E', F') \in R$. It is easy to show that weak bisimulations are closed under union and there exists a largest weak bisimulation which is an equivalence, denoted by $\approx$. States E, F are* weakly bisimilar, *written $E \approx F$, iff there is a weak bisimulation relating them. (Sometimes weak bisimulation is defined with $\Rightarrow$ instead of $\rightarrow$ everywhere. However, the two definitions are equivalent.)*

Weak bisimulation equivalence can also be described by *weak bisimulation games* [13] between two players. One player, the 'attacker', tries to prove that two given processes P_1, P_2 are not weakly bisimilar, while the other player, the 'defender', tries to prevent this. A configuration of a weak bisimulation game is given by a pair of states (A, B). Initially, this pair is (P_1, P_2). The weak bisimulation game is played in rounds. In every round of the game the attacker chooses one process (i.e., either A or B) and performs an action (e.g., $A \xrightarrow{a} A'$). The defender must imitate this move and perform the same action in the other process. However, the defender is allowed to do an arbitrary number of τ-actions before and afterwards, since the defender uses the extended transition relation '$\stackrel{a}{\Rightarrow}$', e.g., $B \stackrel{a}{\Rightarrow} B'$. After the round the new configuration of the weak bisimulation game is (A', B'). If one player cannot move then the other player wins. The defender wins every infinite game. Two processes are weakly bisimilar iff the defender has a winning strategy and non-weakly-bisimilar iff the attacker has a winning strategy.
We show the undecidability of weak bisimulation equivalence for 1-counter nets by a reduction from the acceptance problem of Minsky 2-counter machines.

Definition 3. *A n-counter machine [6] M is described by a finite set of states Q, an initial state $q_0 \in Q$, a final state $accept \in Q$, n counters $c_1, \dots, c_n$ and a finite set of instructions of the form $(q : \; c_i := c_i + 1; \mathsf{goto}\ q')$ or $(q : \; \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ where $i \in \{1, \dots, n\}$ and $q, q', q'' \in Q$. A* configuration *of M is described by a tuple $(q, m_1, \dots, m_n)$ where $q \in Q$ and $m_i \in \mathbb{N}_0$ is the content of the counter c_i ($1 \leq i \leq n$). The possible computation steps are defined as follows:*

1. $(q, m_1, \ldots, m_n) \to (q', m_1, \ldots, m_i + 1, \ldots, m_n)$
   if there is an instruction $(q: \ c_i := c_i + 1; \mathsf{goto}\ q')$.
2. $(q, m_1, \ldots, m_n) \to (q', m_1, \ldots, m_n)$
   if there is an instruction $(q: \ \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ *and* $m_i = 0$.
3. $(q, m_1, \ldots, m_n) \to (q'', m_1, \ldots, m_i - 1, \ldots, m_n)$
   if there is an instruction $(q: \ \mathsf{If}\ c_i = 0\ \mathsf{then\ goto}\ q'\ \mathsf{else}\ c_i := c_i - 1; \mathsf{goto}\ q'')$ *and* $m_i > 0$.

A counter machine is deterministic *iff for every control-state* $q \in Q$ *there is at most one instruction* $(q: \ \ldots\ldots)$ *at this control-state. A deterministic 2-counter machine accepts an input* $n_1, n_2 \in \mathbb{N}_0$ *iff the run starting at configuration* (q_0, n_1, n_2) *is finite and ends in the control-state 'accept'.*

The following problem was shown to be undecidable by Minsky [6].

<u>CM</u>
Instance: A deterministic 2-counter machine M with initial state q_0 and $n_1, n_2 \in \mathbb{N}_0$.
Question: Does M accept (q_0, n_1, n_2) ?

We consider the problem of weak bisimulation equivalence of 1-counter nets.

<u>1-CN $\approx$ 1-CN</u>
Instance: A 1-counter net (S, X, Act, Δ) and $s_1, s_2 \in S$ and $n \in \mathbb{N}_0$.
Question: $s_1 X^n \approx s_2 X^n$?

We show the undecidability of 1-CN $\approx$ 1-CN by a reduction from CM.

3 The Idea

The idea for the reduction is to encode the execution of the 2-counter machine into the weak bisimulation game between the two 1-counter nets. Every computation step of the 2-counter machine is emulated in a finite number of rounds in the weak bisimulation game. The attacker has a universal winning strategy in the weak bisimulation game if and only if the 2-counter machine accepts. Thus, the two 1-counter nets are non-weakly-bisimilar iff the 2-counter machine accepts (i.e., the answer to 1-CN $\approx$ 1-CN is no iff the answer to CM is yes).

The crucial part of the construction is that each of the two 1-counter nets stores the whole configuration of the 2-counter machine. Thus, the two numbers $n_1, n_2 \in \mathbb{N}_0$ in the two counters of the 2-counter machine must be stored in one number n of the 1-counter net. This is done by Gödel-coding of the form $n = 2^{n_1}3^{n_2}$. The problem with this coding is that increment/decrement operations on n_1, n_2 now correspond to multiplication/division operations with constants on n. E.g., $n_2 := n_2 + 1$ is encoded by $n := 3 * n$. Testing n_1 or n_2 for zero is equivalent to testing n for non-divisibility by 2 or 3.

The central part of the proof is to show that it is possible to encode the operations of multiplication and division with constants and tests for divisibility with constants into weak bisimulation games on 1-counter nets. (Note that the same cannot be done for strong bisimulation, and indeed strong bisimilarity is decidable for 1-counter machines and 1-counter nets [3].)

4 Auxiliary Constructions

First, we describe some transition rules used for testing if a given number is exactly twice (or three times) as big an another.

$$\begin{array}{lll} t(1)X \xrightarrow{c} t(1) & t(2)X \xrightarrow{c} t(2)' & t(2)' \xrightarrow{c} t(2) \\ t(3)X \xrightarrow{c} t(3)' & t(3)' \xrightarrow{c} t(3)'' & t(3)'' \xrightarrow{c} t(3) \end{array}$$

Lemma 4. *$t(1)X^n \approx t(1)X^m$ iff $n = m$, $t(1)X^n \approx t(2)X^m$ iff $n = 2m$, and $t(1)X^n \approx t(3)X^m$ iff $n = 3m$.*

Proof. By induction on n, m. □

These transition rules are used for testing if a given number is divisible by 3.

$$\begin{array}{ll} t(3,0)X \xrightarrow{c} t(3,2) & t(3,0)'X \xrightarrow{c} t(3,2)' \\ t(3,2)X \xrightarrow{c} t(3,1) & t(3,2)'X \xrightarrow{c} t(3,1)' \\ t(3,1)X \xrightarrow{c} t(3,0) & t(3,1)'X \xrightarrow{c} t(3,0)' \\ t(3,2) \xrightarrow{d} t(3,2) & t(3,2)'X \xrightarrow{d} t(3,2)'X \\ t(3,1) \xrightarrow{d} t(3,1) & t(3,1)'X \xrightarrow{d} t(3,1)'X \end{array}$$

Lemma 5. $\forall i \in \{0,1,2\}.\ t(3,i)X^n \approx t(3,i)'X^n \iff n \mathit{mod}\, 3 = i$

Proof. By induction on n. For the base case $n = 0$ we have $t(3,0) \not\rightarrow$ and $t(3,0)' \not\rightarrow$ and thus $t(3,0) \approx t(3,0)'$. Furthermore, $t(3,2) \xrightarrow{d} t(3,2)$, but $t(3,2)' \stackrel{d}{\not\rightarrow}$ and thus $t(3,2) \not\approx t(3,2)'$. Finally, $t(3,1) \xrightarrow{d} t(3,1)$, but $t(3,1)' \stackrel{d}{\not\rightarrow}$ and thus $t(3,1) \not\approx t(3,1)'$.
Now let $n \geq 1$. (Let, by convention, $-1 \mathit{mod}\, 3 = 2$.) If $t(3,i)X^n \approx t(3,i)'X^n$ then $t(3,i)X^n \xrightarrow{c} t(3,(i-1) \mathit{mod}\, 3)X^{n-1}$, $t(3,i)'X^n \xrightarrow{c} t(3,(i-1) \mathit{mod}\, 3)'X^{n-1}$ and $t(3,(i-1) \mathit{mod}\, 3)X^{n-1} \approx t(3,(i-1) \mathit{mod}\, 3)'X^{n-1}$. By induction hypothesis $(n-1) \mathit{mod}\, 3 = (i-1) \mathit{mod}\, 3$ and thus $n \mathit{mod}\, 3 = i \mathit{mod}\, 3$. Since $i \in \{0,1,2\}$ we get $n \mathit{mod}\, 3 = i$.
Now let $n \mathit{mod}\, 3 = i$. For $i \in \{1,2\}$ we have $t(3,i)X^n \xrightarrow{d} t(3,i)X^n$ and $t(3,i)'X^n \xrightarrow{d} t(3,i)'X^n$ since $n \geq 1$. (If $i = 0$ then the action d cannot be performed in either process.) Furthermore, we have $t(3,i)X^n \xrightarrow{c} t(3,(i-1) \mathit{mod}\, 3)X^{n-1}$ and $t(3,i)'X^n \xrightarrow{c} t(3,(i-1) \mathit{mod}\, 3)'X^{n-1}$. As $n \mathit{mod}\, 3 = i$ we have $(n-1) \mathit{mod}\, 3 = (i-1) \mathit{mod}\, 3$ and thus by induction hypothesis $t(3,(i-1) \mathit{mod}\, 3)X^{n-1} \approx t(3,(i-1) \mathit{mod}\, 3)'X^{n-1}$. It follows that $t(3,i)X^n \approx t(3,i)'X^n$. □

These transition rules are used for testing if a given number is divisible by 2.

$$\begin{array}{ll} t(2,0)X \xrightarrow{c} t(2,1) & t(2,0)'X \xrightarrow{c} t(2,1)' \\ t(2,1)X \xrightarrow{c} t(2,0) & t(2,1)'X \xrightarrow{c} t(2,0)' \\ t(2,1) \xrightarrow{d} t(2,1) & t(2,1)'X \xrightarrow{d} t(2,1)'X \end{array}$$

Lemma 6. $\forall i \in \{0,1\}.\ t(2,i)X^n \approx t(2,i)'X^n \iff n \mathit{mod}\, 2 = i$

Proof. Analogously to the proof of Lemma 5. □

5 The Main Result

Let M be a Minsky 2-counter machine with a set of control states Q, initial control-state $q_0 \in Q$ and input values $n_1, n_2 \in \mathbb{N}_0$. We construct a 1-counter net (S, X, Act, Δ) such that M accepts (q_0, n_1, n_2) iff $q_0 X^n \not\approx q_0' X^n$, where $n = 2^{n_1} 3^{n_2}$ and $q_0, q_0' \in S$.
The configuration of the 2-counter machine is encoded in the 1-counter net as follows. The control-states of the 2-counter machine are encoded directly into the finite control of the 1-counter net. The natural numbers $n_1, n_2 \in \mathbb{N}_0$ stored in the counters c_1, c_2 of the 2-counter machine are encoded as X^n in the 1-counter net, where $n = 2^{n_1} 3^{n_2}$.

Remark 7. Let $n_1, n_2 \in \mathbb{N}_0$ and $n = 2^{n_1} 3^{n_2}$. Then we have: $n_1 = 0$ iff $n \mathit{ mod } 2 \neq 0$, $n_2 = 0$ iff $n \mathit{ mod } 3 \neq 0$. $n_1 := n_1 + 1$ corresponds to $n := 2n$, $n_1 := n_1 - 1$ to $n := n/2$, $n_2 := n_2 + 1$ to $n := 3n$ and $n_2 := n_2 - 1$ to $n := n/3$.

For every instruction $(q: \;\; c_2 := c_2 + 1; \mathtt{goto}\ p)$ of the Minsky 2-counter machine we define the following rules of the 1-counter net. For some control-states q we define a new control-state $G(q)$ ('G' for 'generate') which behaves as follows: First an arbitrary number of symbols X are added or removed using only τ-actions. Then the action 'a' is performed and the control-state becomes q.

$$\begin{array}{ll}
G(q_2) \xrightarrow{\tau} G(q_2)X & G(q_1') \xrightarrow{\tau} G(q_1')X \\
G(q_2)X \xrightarrow{\tau} G(q_2) & G(q_1')X \xrightarrow{\tau} G(q_1') \\
G(q_2) \xrightarrow{a} q_2 & G(q_1') \xrightarrow{a} q_1'
\end{array}$$

Now come the rules for encoding the operation.

$$\begin{array}{ll}
q \xrightarrow{a} q_1 & \\
q \xrightarrow{\tau} G(q_1') & q' \xrightarrow{\tau} G(q_1') \\
q_1 \xrightarrow{t} t(3) & q_1' \xrightarrow{t} t(1) \\
 & q_1' \xrightarrow{a} q_2' \\
q_1 \xrightarrow{\tau} G(q_2) & q_1' \xrightarrow{\tau} G(q_2) \\
q_2 \xrightarrow{t} t(1) & q_2' \xrightarrow{t} t(1) \\
q_2 \xrightarrow{a} p & q_2' \xrightarrow{a} p'
\end{array}$$

The following lemma shows that these rules encode the 2-counter machine operation into the weak bisimulation game on the 1-counter net, provided that both players (attacker and defender) play optimally (i.e., avoid to lose if they can).

Lemma 8. *Let $(q: \; c_2 := c_2 + 1; \mathtt{goto}\ p)$ be the instruction of the Minsky 2-counter machine at control-state q. Let $n_1, n_2 \in \mathbb{N}_0$ and $n = 2^{n_1} 3^{n_2}$. The weak bisimulation game starting at the configuration $(qX^n, q'X^n)$ has the following properties:*

- *The attacker has a strategy by which he can either (depending on the moves of the defender player) win, or at least force the weak bisimulation game into the configuration $(pX^m, p'X^m)$, where $m = 2^{n_1} 3^{n_2+1}$.*
- *The defender has a strategy by which he can either (depending on the moves of the attacker) win, or at least enforce that the weak bisimulation game goes on and eventually reaches the configuration $(pX^m, p'X^m)$, where $m = 2^{n_1} 3^{n_2+1}$.*

Proof. We start the weak bisimulation game at $(qX^n, q'X^n)$. It proceeds as follows:

1. The attacker must play $qX^n \xrightarrow{a} q_1X^n$, otherwise the defender could make the two processes syntactically equal in the same round and win.
2. Now the defender has several choices:
   (a) If the defender plays $q'X^n \stackrel{a}{\Rightarrow} G(q_2)X^k$ for some $k \in \mathbb{N}_0$ (by the rules for $G(q_1')$ and $G(q_2)$) then the attacker can win. This is because $q_1X^n \not\approx G(q_2)X^k$, since $q_1X^n \xrightarrow{t}$ and $G(q_2)X^k \stackrel{t}{\not\Rightarrow}$.
   (b) Therefore the defender will play $q'X^n \stackrel{a}{\Rightarrow} q_1'X^k$ for some $k \in \mathbb{N}_0$ (by definition of the rules for $G(q_1')$). The defender will choose $k = 3n$, because otherwise the attacker could win. If $k \neq 3n$ then the attacker can play $q_1X^n \xrightarrow{t} t(3)X^n$ to which the defender must reply by $q_1'X^k \xrightarrow{t} t(1)X^k$. By Lemma 4 we have $t(3)X^n \not\approx t(1)X^k$ for $k \neq 3n$ and so the attacker can win. Therefore the defender will choose $k = 3n$. The configuration of the weak bisimulation game is now $(q_1X^n, q_1'X^k)$ for $k = 3n$.
3. Now the attacker has several choices:
   (a) If the attacker plays $q_1X^n \xrightarrow{t} t(3)X^n$ the defender replies $q_1'X^k \xrightarrow{t} t(1)X^k$. Since $k = 3n$ we have $t(3)X^n \approx t(1)X^k$ by Lemma 4 and thus the defender can win. (Analogously if the attacker plays $q_1'X^k \xrightarrow{t} t(1)X^k$.)
   (b) If the attacker plays $q_1X^n \xrightarrow{\tau} G(q_2)X^n$ the defender replies $q_1'X^k \stackrel{\tau}{\Rightarrow} G(q_2)X^n$ (by definition of the rules for $G(q_2)$) and the defender wins. Similarly, if the attacker plays $q_1'X^k \xrightarrow{\tau} G(q_2)X^k$ then the defender replies $q_1X^n \stackrel{\tau}{\Rightarrow} G(q_2)X^k$.
   (c) Therefore the attacker will choose the only remaining move $q_1'X^k \xrightarrow{a} q_2'X^k$.
4. To this move the defender will by reply by $q_1X^n \stackrel{a}{\Rightarrow} q_2X^l$ (by definition of the rules for $G(q_2)$) for some $l \in \mathbb{N}_0$. The defender will choose $l = k$, because otherwise the attacker could win. If $l \neq k$ then the attacker can play $q_2X^l \xrightarrow{t} t(1)X^l$ to which the defender must reply by $q_2'X^k \xrightarrow{t} t(1)X^k$. By Lemma 4 we have $t(1)X^l \not\approx t(1)X^k$ for $l \neq k$ and so the attacker can win. Therefore the defender will choose $l = k$. The configuration of the weak bisimulation game is now $(q_2X^k, q_2'X^k)$ with $k = 3n$.
5. Now the attacker has two choices:
   (a) If the attacker plays $q_2X^k \xrightarrow{t} t(1)X^k$ the defender replies $q_2'X^k \xrightarrow{t} t(1)X^k$ and the defender wins. (Analogously if the attacker plays $q_2'X^k \xrightarrow{t} t(1)X^k$.)
   (b) Therefore the attacker will play $q_2X^k \xrightarrow{a} pX^k$ and the defender replies $q_2'X^k \xrightarrow{a} p'X^k$ (or vice-versa).
6. So, finally, unless one of the players has played sub-optimally and lost, we have reached the configuration $(pX^k, p'X^k)$ with $k = 3n$. As $n = 2^{n_1}3^{n_2}$ we have $k = 2^{n_1}3^{n_2+1} = m$. □

Remark 9. For the instructions of the form $(q:\ c_1 := c_1 + 1; \mathsf{goto}\ p)$ the construction is analogous. The only difference is that the constant $t(3)$ is replaced by $t(2)$ (since the number n in the 1-counter net is not tripled, but doubled). A lemma analogous to Lemma 8 is easy to show.

For every instruction of the form $(q:\ \mathsf{If}\ c_2 = 0\ \mathsf{then\ goto}\ p\ \mathsf{else}\ c_2 := c_2 - 1; \mathsf{goto}\ r)$ of the Minsky 2-counter machine we define the following rules of the 1-counter net.

First, some auxiliary rules can generate (or remove) an arbitrary number of symbols X.

$$
\begin{array}{ll}
G(D(3,r)'_1) \xrightarrow{\tau} G(D(3,r)'_1)X & G(D(3,r)_2) \xrightarrow{\tau} G(D(3,r)_2)X \\
G(D(3,r)'_1)X \xrightarrow{\tau} G(D(3,r)'_1) & G(D(3,r)_2)X \xrightarrow{\tau} G(D(3,r)_2) \\
G(D(3,r)'_1) \xrightarrow{a} D(3,r)'_1 & G(D(3,r)_2) \xrightarrow{a} D(3,r)_2
\end{array}
$$

The following auxiliary rules are used to encode the operation of dividing the number n by 3 (provided that it is divisible by 3) and then going to control-state r or r'.

$$
\begin{array}{ll}
D(3,r) \xrightarrow{a} D(3,r)_1 & \\
D(3,r) \xrightarrow{\tau} G(D(3,r)'_1) & D(3,r)' \xrightarrow{\tau} G(D(3,r)'_1) \\
D(3,r)_1 \xrightarrow{t} t(1) & D(3,r)' \xrightarrow{t} t(3) \\
 & D(3,r)'_1 \xrightarrow{a} D(3,r)'_2 \\
D(3,r)_1 \xrightarrow{\tau} G(D(3,r)_2) & D(3,r)'_1 \xrightarrow{\tau} G(D(3,r)_2) \\
D(3,r)_2 \xrightarrow{t} t(1) & D(3,r)'_2 \xrightarrow{t} t(1) \\
D(3,r)_2 \xrightarrow{a} r & D(3,r)'_2 \xrightarrow{a} r'
\end{array}
$$

Now come the rules for encoding the test-and-decrement operation. The intuition is that by action 'a' the attacker claims that $c_2 = 0$, (i.e., $n \; mod \; 3 \neq 0$) and by action 'b' the attacker claims $c_2 > 0$ (i.e., $n \; mod \; 3 = 0$). The defender can check these claims and win if they are wrong. The states $q_1(i)$ or $q_2(i)$ encode the counter-claims of the defender that $n \; mod \; 3 = i$.

$$
\begin{array}{llll}
q \xrightarrow{a} q_1 & q \xrightarrow{b} q_2 & & \\
q \xrightarrow{a} q'_1 & q \xrightarrow{b} q'_2 & q' \xrightarrow{a} q'_1 & q' \xrightarrow{b} q'_2 \\
q \xrightarrow{a} q_1(0) & q \xrightarrow{b} q_2(1) & q' \xrightarrow{a} q_1(0) & q' \xrightarrow{b} q_2(1) \\
 & q \xrightarrow{b} q_2(2) & & q' \xrightarrow{b} q_2(2) \\
q_1 \xrightarrow{a} p & & q'_1 \xrightarrow{a} p' & q_1(0) \xrightarrow{a} p \\
q_1 \xrightarrow{t} t(3,0) & & q'_1 \xrightarrow{t} t(3,0) & q_1(0) \xrightarrow{t} t(3,0)' \\
q_2 \xrightarrow{a} D(3,r) & & q'_2 \xrightarrow{a} D(3,r)' & q_2(1) \xrightarrow{a} D(3,r) \\
 & & & q_2(2) \xrightarrow{a} D(3,r) \\
q_2 \xrightarrow{t_1} t(3,1) & q_2 \xrightarrow{t_2} t(3,2) & q'_2 \xrightarrow{t_1} t(3,1) & q'_2 \xrightarrow{t_2} t(3,2) \\
 & & q_2(1) \xrightarrow{t_1} t(3,1)' & q_2(1) \xrightarrow{t_2} t(3,2) \\
 & & q_2(2) \xrightarrow{t_1} t(3,1) & q_2(2) \xrightarrow{t_2} t(3,2)'
\end{array}
$$

Lemma 10. *If $k = n \; mod \; 3$ and $k \in \{1,2\}$ then $q_2X^n \approx q_2(k)X^n$.*

Proof. Let $j := 3 - k$. The defender has the following winning strategy:

- If the attacker plays $q_2X^n \xrightarrow{a} D(3,r)X^n$ then the defender can play $q_2(k)X^n \xrightarrow{a} D(3,r)X^n$ (and vice-versa) and thus the defender wins.
- If the attacker plays $q_2X^n \xrightarrow{t_j} t(3,j)X^n$ then the defender can play $q_2(k)X^n \xrightarrow{t_j} t(3,j)X^n$ (and vice-versa) and thus the defender wins.
- If the attacker plays $q_2X^n \xrightarrow{t_k} t(3,k)X^n$ then the defender can play $q_2(k)X^n \xrightarrow{t_k} t(3,k)'X^n$ (and vice-versa). By Lemma 5 $t(3,k)X^n \approx t(3,k)'X^n$ (since $k = n \; mod \; 3$) and thus the defender wins. $\square$

Lemma 11. *If $n \bmod 3 = 0$ then $q_1 X^n \approx q_1(0) X^n$.*

Proof. The defender has the following winning strategy.

- If the attacker plays $q_1 X^n \overset{a}{\to} pX^n$ then the defender replies $q_1(0)X^n \overset{a}{\to} pX^n$ (and vice-versa) and thus the defender wins.
- If the attacker plays $q_1 X^n \overset{t}{\to} t(3,0)X^n$ then the defender replies $q_1(0)X^n \overset{t}{\to} t(3,0)'X^n$ (and vice-versa). By Lemma 5 we have $t(3,0)X^n \approx t(3,0)'X^n$ (since $n \bmod 3 = 0$) and thus the defender wins. □

The following two lemmas show that these rules encode the 2-counter machine operation test-and-decrement into the weak bisimulation game on the 1-counter net.

Lemma 12. *Let $(q : \textsf{If } c_2 = 0 \textsf{ then goto } p \textsf{ else } c_2 := c_2 - 1; \textsf{goto } r)$ be a Minsky 2-counter machine instruction, $n_1, n_2 \in \mathbb{N}_0$ and $n = 2^{n_1}3^{n_2}$. If $n_2 = 0$ then the weak bisimulation game starting at the configuration $(qX^n, q'X^n)$ has the following properties:*

- *The attacker has a strategy by which he can either (depending on the moves of the defender player) win, or at least force the weak bisimulation game into the configuration $(pX^n, p'X^n)$.*
- *The defender has a strategy by which he can either (depending on the moves of the attacker) win, or at least enforce that the weak bisimulation game goes on and eventually reaches the configuration $(pX^n, p'X^n)$.*

Proof. Note that $n \bmod 3 \in \{1, 2\}$, because $n_2 = 0$. We start the weak bisimulation game at $(qX^n, q'X^n)$. It proceeds as follows:

1. The attacker has several choices:
   (a) If the attacker makes any move that is not $qX^n \overset{a}{\to} q_1 X^n$ or $qX^n \overset{b}{\to} q_2 X^n$ then the defender can make the two processes syntactically equal in the same round and thus the defender wins.
   (b) If the attacker plays $qX^n \overset{b}{\to} q_2 X^n$ then the defender has the following strategy to win. Let $k := n \bmod 3$. $k \in \{1, 2\}$, since $n_2 = 0$. The defender plays $q'X^n \overset{a}{\to} q_2(k)X^n$. By Lemma 10 $q_2 X^n \approx q_2(k)X^n$ and the defender wins.
   (c) Therefore the attacker will do $qX^n \overset{a}{\to} q_1 X^n$.
2. Now the defender has two choices:
   (a) If the defender plays $q'X^n \overset{a}{\to} q_1(0)X^n$ then the attacker has the following winning strategy. In the next round the attacker plays $q_1 X^n \overset{t}{\to} t(3,0)X^n$ to which the defender can only reply $q_1(0)X^n \overset{t}{\to} t(3,0)'X^n$. By Lemma 5 we have $t(3,0)X^n \not\approx t(3,0)'X^n$, because $n \bmod 3 \neq 0$, and thus the attacker can win.
   (b) Therefore the defender plays $q'X^n \overset{a}{\to} q_1' X^n$. The configuration is now $(q_1 X^n, q_1' X^n)$.
3. Now the attacker has two choices: If the attacker plays $q_1 X^n \overset{t}{\to} t(3,0)X^n$ then the defender replies $q_1' X^n \overset{t}{\to} t(3,0)X^n$ (and vice versa) and so the defender wins. Therefore the attacker will play $q_1 X^n \overset{a}{\to} pX^n$ (or $q_1' X^n \overset{a}{\to} p'X^n$, this case is symmetric).
4. The defender can only reply $q_1' X^n \overset{a}{\to} p'X^n$. The configuration is now $(pX^n, p'X^n)$.

□

Lemma 13. *Let* $(q: \mathsf{If}\ c_2 = 0\ \mathsf{then\ goto}\ p\ \mathsf{else}\ c_2 := c_2 - 1; \mathsf{goto}\ r)$ *be the instruction of the Minsky 2-counter machine at control-state* q. *Let* $n_1, n_2 \in \mathbb{N}_0$ *and* $n = 2^{n_1}3^{n_2}$. *If* $n_2 > 0$ *then the weak bisimulation game starting at the configuration* $(qX^n, q'X^n)$ *has the following properties:*

- *The attacker has a strategy by which he can either (depending on the moves of the defender player) win, or at least force the weak bisimulation game into the configuration* $(rX^m, r'X^m)$, *where* $m = 2^{n_1}3^{n_2-1}$.
- *The defender has a strategy by which he can either (depending on the moves of the attacker) win, or at least enforce that the weak bisimulation game goes on and eventually reaches the configuration* $(rX^m, r'X^m)$, *where* $m = 2^{n_1}3^{n_2-1}$.

Proof. Note that $n\ mod\ 3 = 0$, because $n_2 > 0$. We start the weak bisimulation game at $(qX^n, q'X^n)$. It proceeds as follows:

1. The attacker has several choices:
   (a) If the attacker makes any move that is not $qX^n \xrightarrow{a} q_1X^n$ or $qX^n \xrightarrow{b} q_2X^n$ then the defender can make the two processes syntactically equal in the same round and thus the defender wins.
   (b) If the attacker plays $qX^n \xrightarrow{a} q_1X^n$ then the defender can reply $q'X^n \xrightarrow{a} q_1(0)X^n$ and wins, because $q_1X^n \approx q_1(0)X^n$ for $n\ mod\ 3 = 0$, by Lemma 11.
   (c) Therefore the attacker will play $qX^n \xrightarrow{b} q_2X^n$.
2. Now the defender has three choices:
   (a) If the defender does $q'X^n \xrightarrow{b} q_2(1)X^n$ then the attacker has the following winning strategy. The attacker plays $q_2X^n \xrightarrow{t_1} t(3,1)X^n$ to which the defender can only reply $q_2(1)X^n \xrightarrow{t_1} t(3,1)'X^n$. By Lemma 5 $t(3,1)X^n \not\approx t(3,1)'X^n$ (since $n\ mod\ 3 \neq 1$) and thus the attacker can win.
   (b) Similarly, if the defender does $q'X^n \xrightarrow{b} q_2(2)X^n$ then the attacker can also win by Lemma 5, since $n\ mod\ 3 \neq 2$.
   (c) Therefore the defender plays $q'X^n \xrightarrow{b} q_2'X^n$. The configuration is now $(q_2X^n, q_2'X^n)$.
3. The attacker now plays $q_2X^n \xrightarrow{a} D(3,r)X^n$ to which the defender replies $q_2'X^n \xrightarrow{a} D(3,r)'X^n$ (or vice-versa). (If the attacker played any other move by action t_1 or t_2 then the processes would become syntactically equal in the same round and the defender would win.) The configuration is now $(D(3,r)X^n, D(3,r)'X^n)$.
4. The attacker must now play $D(3,r)X^n \xrightarrow{a} D(3,r)_1X^n$, otherwise the defender could make the two processes syntactically equal in the same round and win.
5. Now the defender has several choices:
   (a) If the defender plays $D(3,r)'X^n \xrightarrow{a} G(D(3,r)_2)X^k$ for some $k \in \mathbb{N}_0$ (by the rules for $G(D(3,r)_1')$ and $G(D(3,r)_2)$) then the attacker wins. This is because $D(3,r)_1X^n \not\approx G(D(3,r)_2)X^k$, since $D(3,r)_1X^n \xrightarrow{t}$ and $G(D(3,r)_2)X^k \stackrel{t}{\not\Rightarrow}$.
   (b) Therefore the defender plays $D(3,r)'X^n \stackrel{a}{\Rightarrow} D(3,r)_1'X^k$ for some $k \in \mathbb{N}_0$ (by def. of the rules for $G(D(3,r)_1')$). The defender will choose $k = n/3$ (this is possible, since $n\ mod\ 3 = 0$) for the following reason. If $k \neq n/3$ then the attacker can play $D(3,r)_1X^n \xrightarrow{t} t(1)X^n$ to which the defender must reply by $D(3,r)_1'X^k \xrightarrow{t} t(3)X^k$. By Lemma 4 we have $t(1)X^n \not\approx t(3)X^k$ for $k \neq n/3$ and so the attacker can win. Therefore the defender will choose $k = n/3$. So the configuration is now $(D(3,r)_1X^n, D(3,r)_1'X^k)$ with $k = n/3$.

6. Now the attacker has several choices:
   (a) If the attacker plays $D(3,r)_1X^n \xrightarrow{t} t(1)X^n$ the defender replies $D(3,r)'_1X^k \xrightarrow{t} t(3)X^k$. Since $k = n/3$ we have $t(1)X^n \approx t(3)X^k$ by Lemma 4 and thus the defender can win. (Analogously if the attacker plays $D(3,r)'_1X^k \xrightarrow{t} t(3)X^k$.)
   (b) If the attacker plays $D(3,r)_1X^n \xrightarrow{\tau} G(D(3,r)_2)X^n$ then the defender can reply $D(3,r)'_1X^k \stackrel{\tau}{\Rightarrow} G(D(3,r)_2)X^n$ (by def. of the rules for $G(D(3,r)_2)$) and the defender wins. Similarly, if the attacker plays $D(3,r)'_1X^k \xrightarrow{\tau} G(D(3,r)_2)X^k$ then the defender replies $D(3,r)_1X^n \stackrel{\tau}{\Rightarrow} G(D(3,r)_2)X^k$.
   (c) Therefore the attacker chooses the only remaining move $D(3,r)'_1X^k \xrightarrow{a} D(3,r)'_2X^k$.
7. To this move the defender will by reply by $D(3,r)_1X^n \stackrel{a}{\Rightarrow} D(3,r)_2X^l$ (by definition of the rules for $G(D(3,r)_2)$) for some $l \in \mathbb{N}_0$. The defender will choose $l = k$ for the following reason. If $l \neq k$ then the attacker can play $D(3,r)_2X^l \xrightarrow{t} t(1)X^l$ to which the defender must reply by $D(3,r)'_2X^k \xrightarrow{t} t(1)X^k$. By Lemma 4 we have $t(1)X^l \not\approx t(1)X^k$ for $l \neq k$ and so the attacker can win. Therefore the defender will choose $l = k$. So the configuration of the weak bisimulation game is now $(D(3,r)_2X^k, D(3,r)'_2X^k)$ with $k = n/3$.
8. The attacker has two choices:
   (a) If the attacker plays $D(3,r)_2X^k \xrightarrow{t} t(1)X^k$ the defender replies $D(3,r)'_2X^k \xrightarrow{t} t(1)X^k$ (and vice-versa) and the defender wins.
   (b) Therefore the attacker will play $D(3,r)_2X^k \xrightarrow{a} rX^k$ and the defender must reply $D(3,r)'_2X^k \xrightarrow{a} r'X^k$ (or vice-versa).
9. So, finally, unless one of the players has played sub-optimally and lost, we have reached the configuration $(rX^k, r'X^k)$ with $k = n/3$. As $n = 2^{n_1}3^{n_2}$ and $n_2 > 0$ we have $k = 2^{n_1}3^{n_2-1} = m$. □

Remark 14. For the instructions of the form (q : If $c_1 = 0$ then goto p else $c_1 := c_1 - 1$; goto r) the construction is similar, but slightly simpler. In this case we test the number n for divisibility by 2 and divide by 2 if possible (instead of by 3 for instructions on counter c_2 as shown above). Lemmas analogous to Lemma 12 and Lemma 13 are easy to show.

Finally, we add one last rule to the 1-counter net, which is used to distinguish accepting- and non-accepting states of the Minsky 2-counter machine. By Definition 3 the 2-counter machine has only one accepting state '*accept*'. We add the rule

$$accept \xrightarrow{e} accept$$

Thus we get $acceptX^n \not\approx accept'X^m$ for all n, m, since action e is not possible at control-state $accept'$.

Lemma 15. *Let M be a Minsky 2-counter machine with a set of control states Q, initial control-state $q_0 \in Q$ and input values $n_1, n_2 \in \mathbb{N}_0$. One can effectively construct a 1-counter net (S, X, Act, Δ) (depending only on M) such that M accepts (q_0, n_1, n_2) iff $q_0X^n \not\approx q'_0X^n$, where $n = 2^{n_1}3^{n_2}$ and $q_0, q'_0 \in S$.*

Proof. The 1-counter net is constructed as shown above and depends only on M. Now we show the correctness.

- If M accepts (q_0, n_1, n_2) then the attacker has a winning strategy in the weak bisimulation game starting at $(q_0 X^n, q'_0 X^n)$. By Lemma 8, Remark 9, Lemma 12, Lemma 13 and Remark 14 the attacker has a strategy by which (depending on the moves of the defender) he either wins directly, or simulates the 2-counter machine operation correctly. Since M accepts (q_0, n_1, n_2), the weak bisimulation game will eventually reach some configuration $(\mathit{accept} X^i, \mathit{accept}' X^i)$ for some i (unless the attacker has already won earlier). Now the attacker wins, because $\mathit{accept} X^i \not\approx \mathit{accept}' X^i$. Therefore $q_0 X^n \not\approx q'_0 X^n$.
- If M does not accept (q_0, n_1, n_2) then the defender has a winning strategy in the weak bisimulation game starting at the configuration $(q_0 X^n, q'_0 X^n)$. By Lemma 8, Remark 9, Lemma 12, Lemma 13 and Remark 14 the defender has a strategy by which (depending on the moves of the attacker) he either wins directly, or simulates the 2-counter machine operation correctly. Since M does not accept (q_0, n_1, n_2), the weak bisimulation game will never reach a configuration with the state accept. So the weak bisimulation game will either continue forever, or the defender will win directly by one of the cases shown above. In any case the defender wins and thus $q_0 X^n \approx q'_0 X^n$. □

Now we can show the main theorem.

Theorem 16. *There exists a fixed 1-counter net, such that, for input $n \in \mathbb{N}_0$, the question $q_0 X^n \approx q'_0 X^n$ is undecidable.*

Proof. There exists a fixed universal Minsky 2-counter machine M_u (analogously to the universal Turing-machine) [6] for which the problem, for input $m \in \mathbb{N}_0$, if it accepts $(q_0, m, 0)$ is undecidable. Based on M_u, we construct a fixed 1-counter net as shown above. Let $n = 2^m$. By Lemma 15 we have $q_0 X^n \not\approx q'_0 X^n$ iff M_u accepts $(q_0, m, 0)$. □

Corollary 17. *Weak bisimulation equivalence is undecidable for 1-counter nets and 1-counter machines.*

Remark 18. The question $q_0 X^0 \approx q'_0 X^0$ is also undecidable for general (non-fixed) 1-counter nets, since acceptance of $(q_0, 0, 0)$ is undecidable for 2-counter machines.
Furthermore, all our undecidability results carry over to the subclass of *normed* 1-counter nets. (Normedness means here that from every reachable configuration it is possible to empty the unbounded place.) It suffices to add the following rules to our construction above: $q \xrightarrow{y} z$, $q' \xrightarrow{y} z$ and $zX \xrightarrow{\tau} z$ for all $q \in Q$. (z is a new state and y a new action.) This modified system is normed, but still preserves all properties shown above.

6 Conclusion

We have shown the undecidability of weak bisimulation equivalence for (normed) 1-counter nets and 1-counter machines. (This contrasts with strong bisimulation equivalence which is decidable for these models [3]). Our undecidability result is more general than Srba's result on undecidability of weak bisimilarity for pushdown automata [12]. No stack is needed, since one simple counter suffices. Moreover, it is not even necessary that the counter can be tested for zero. Our construction uses only one unbounded Petri net place.

However, a crucial requirement for our construction is the existence of a global finite control of our (possibly infinite-state) processes. Therefore our undecidability proof does not carry over to those classes of infinite-state processes that are not closed under product with finite automata, like context-free processes (BPA), basic parallel processes (BPP) or PA-processes (see, e.g., [4, 10]). Undecidability of weak bisimilarity for PA-processes has been shown with a very different technique [11]. Decidability of weak bisimilarity for BPA and BPP is still open, but conjectured to be decidable (especially for BPP, due to some recent results by Jančar [10]).
Still, (normed) 1-counter nets are a very weak model that is subsumed by most classes of infinite-state systems. Therefore one can say as a rule of thumb: “Weak bisimilarity is undecidable for most classes of infinite-state systems that are closed under product with finite-automata.”

Remark 19. The undecidability result for weak bisimilarity of 1-counter nets carries over to the even weaker model of lossy 1-counter nets (where the unbounded place can spontaneously lose tokens). Note that, up-to weak bisimilarity, lossy 1-counter nets are a proper subclass of 1-counter nets. The proof is similar to the one given here, but more technically complex in some details. The idea is to use an additional technique from [8] by which one can ensure that whenever one player loses tokens then the other player wins, thus effectively ruling out lossy behavior. The proof can be found in the forthcoming journal version of this paper.

References

[1] O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification on infinite structures. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, chapter 9, pages 545–623. Elsevier Science, 2001.
[2] P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148:281–301, 1995.
[3] P. Jančar. Decidability of bisimilarity for one-counter processes. *Information and Computation*, 158:1–17, 2000.
[4] R. Mayr. Process rewrite systems. *Information and Computation*, 156(1):264–286, 2000.
[5] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
[6] M.L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.
[7] J.L. Peterson. *Petri net theory and the modeling of systems*. Prentice-Hall, 1981.
[8] Ph. Schnoebelen. Bisimulation and other undecidable equivalences for lossy channel systems. In *Proc. of TACS 2001*, volume 2215 of *LNCS*, pages 385–399. Springer Verlag, 2001.
[9] G. Sénizergues. Decidability of bisimulation equivalence for equational graphs of finite outdegree. In *Proc. of FOCS'98*. IEEE, 1998.
[10] J. Srba. Roadmap of infinite results. *Bulletin of the European Association for Theoretical Computer Science*, 78:163–175, October 2002. Columns: Concurrency. Regularly updated online version at `http://www.brics.dk/~srba/roadmap`.
[11] J. Srba. Undecidability of weak bisimilarity for PA-processes. In *Proceedings of 6th International Conference on Developments in Languague Theory (DLT'02)*, LNCS. Springer-Verlag, 2002. To appear.
[12] J. Srba. Undecidability of weak bisimilarity for pushdown processes. In *Proc. of CONCUR 2002*, volume 2421 of *LNCS*, pages 579–593. Springer Verlag, 2002.
[13] C. Stirling. The joys of bisimulation. In *Proc. of MFCS'98*, volume 1450 of *LNCS*, pages 142–151. Springer Verlag, 1998.