Undecidability of Weak Bisimulation Equivalence in Unary One-Counter Petri Nets

Jana Hofmann

Master of Science
Computer Science
School of Informatics
University of Edinburgh

2017

Abstract

In this thesis, we present a transformation to remove labels from one-counter Petri nets and prove that it preserves weak bisimilarity of two states in the system. We then apply this construction in a reduction to show that weak bisimulation equivalence is undecidable in unary one-counter Petri nets. As an addition, we also discuss lossyness in one-counter Petri nets. We compare possible definitions and comment on an approach to prove that weak bisimilarity could also be undecidable in lossy one-counter Petri nets.

Acknowledgements

First of all, I would like to thank my supervisor Dr. Richard Mayr for providing me with this challenging topic. His valuable ideas and advice had a great impact on the thesis. I especially appreciate the deeper insight he gave me during many meetings and written conversations.

Furthermore, I would like to express my gratitude to the *Cusanuswerk* for (financially) supporting my studies and the year abroad.

My fellow students and friends made my stay in Edinburgh such a lovely experience. Special thanks go to Lukas for sharing my interest in theoretical computer science, Chris for his encouragement and helpful comments and my flatmates, Judy and Shubha, for all the spontaneous late-night dinners.

Lastly, I would like to thank my family for always believing in me.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Jana Hofmann*)

Contents

Chapter 1

Introduction

Modelling various kinds of systems has always been an important aspect in Computer Science. Finding suitable abstractions is a major step in order to reliably verify the correctness of a system and to prove desirable properties. Over the years, a wide range of models has evolved which all capture different aspects of real-world systems. Although many of them have been studied extensively, there are still various open problems. Models that are considerably complex quickly become Turing powerful so that it is clear that many interesting questions like termination are undecidable. For weaker models, these kinds of problems become more interesting.

1.1 Petri Nets

Petri nets [8] belong to the area of process models and are not Turing powerful. They model a concurrent, flow-like behaviour as e.g. in chemical processes. Petri nets consist of states (so-called *places*) which contain *tokens* and transition rules that describe how the tokens move between the places. A sample Petri net is given in Fig. 1.1. It consists of three places and two transitions which are labelled with t_1 and t_2. The two places on the left both contain a token initially. The *multiplicity* on the arcs describes the number of tokens a transition consumes and produces when it *fires*.

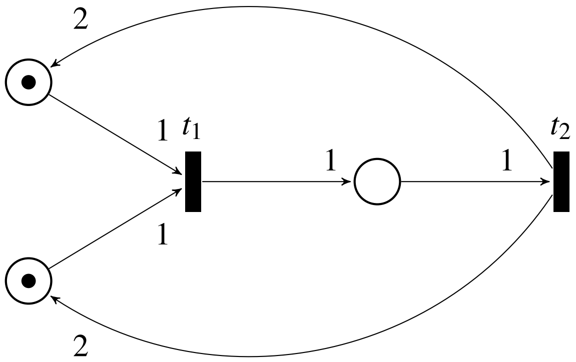

Figure 1.1: An Example Petri Net

To see how the tokens move through the net, see Fig. 1.2. In this net, all places can contain an arbitrary number of tokens, depending on the execution of the net.

1.2 Aim of the Thesis

One of the most important questions when modelling systems is whether two configurations in a model behave equivalently. For state-based models, the best known behavioural equivalence is bisimulation equivalence. While being stronger than path equivalence but weaker than simple isomorphism, it is also a congruence. These properties make bisimulation very convenient to work with. Besides normal (also called strong) bisimilarity, there is also weak bisimilarity which allows to study systems modular internal, invisible actions. Weak bisimilarity is known to be undecidable for many, even comparatively weak systems (see Chapter 2).

In this thesis, we investigate the decidability of weak bisimulation equivalence in unary one-counter Petri nets. Unary one-counter nets are a very restricted version of Petri nets. They only have one unbounded place (i.e. all other places do not contain more than a fixed number of tokens). Furthermore, only a single action is allowed for transition labels. Unary one-counter nets can thus be treated as being unlabelled. We claim that weak bisimulation equivalence is still undecidable for this kind of Petri nets. Con-

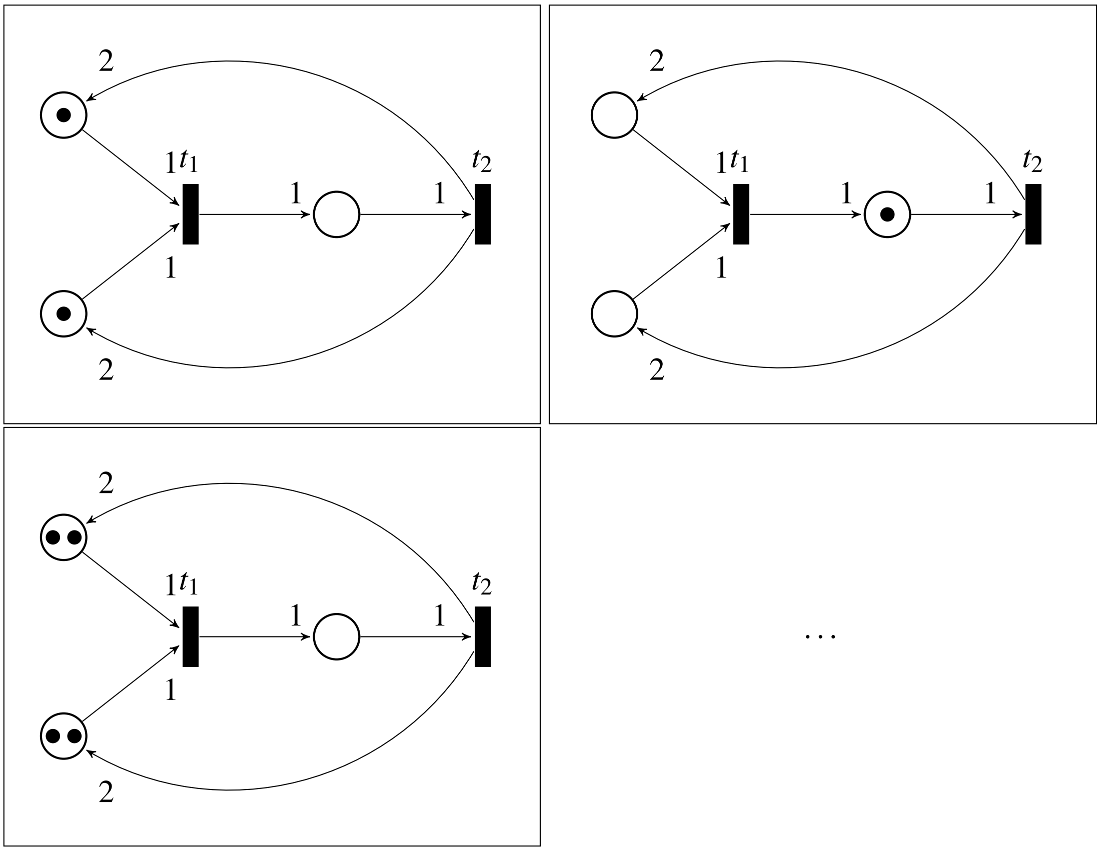

Figure 1.2: An Execution in a Petri Net

sidering that unlabelled one-counter nets are one of the weakest state-based systems which still have a reasonable expressiveness, this is a very strong result. It makes us conclude that weak bisimilarity seems to be undecidable for a wide range of systems.

This thesis is structured as follows. We first give a survey on related literature to put our work into context. We then state important definitions and lemmas that we build this thesis on. In Chapter 4, we give a transformation from one-counter nets to unlabelled one-counter nets. This transformation is used in a reduction from the problem of weak bisimilarity in one-counter nets to weak bisimilarity in unlabelled one-counter nets to formally prove our claim. Subsequently, we discuss another possibility to weaken one-counter nets: Lossy one-counter nets allow the system to non-deterministically lose data. We expected to be able to prove that weak bisimilarity is also undecidable in lossy one-counter nets using a straightforward argument. However, it turned out that this problem is much more complicated than we initially thought. In Chapter 5, we therefore discuss a few obstacles and challenges we encountered when trying to give an undecidability proof. Finally, we evaluate our work and give a conclusion.

Chapter 2

Background

In this chapter, we introduce systems that are related to Petri nets and give a survey over results in literature that are of interest for our work. We mainly concentrate on decidability results for strong and weak bisimulation equivalence.

Philosophically, Petri nets seem to be very different from models for computation as e.g. register machines. Mathematically, they are surprisingly similar. A related system are counter machines. Counter machines are simple register machines where the registers contain a single integer value that can be increased, decreased and tested for zero by instructions.

Petri nets are mathematically equivalent to the subclass of counter machines where the counters can not explicitly be tested for zero. These are also called vector addition systems with states (VASS). Petri nets only allow implicit tests for zero meaning that a transition can be restricted to the case where a place contains at least a fixed number of tokens. Since one-counter nets are the subclass of Petri nets where there is a finite number of control states and only one unbounded place, one-counter nets are a subclass of one-counter machines.

The concept of lossyness in counter machines has been introduced by Mayr in [5] as a version of Minsky counter machines [7]. In lossy counter machines, the counters can decrease spontaneously. Lossyness is also interesting for other systems, e.g. channel systems where messages can get lost in unreliable channels. It might be surprising that lossyness actually weakens a system. This is due to the fact that neither the machine nor an observer is able to control or even notice when and what data is lost.

In Table 2.1, you can find a survey of decidability results of strong and weak bisimulation equivalence in systems that are of interest for us. Note that we do not consider equivalences between different kinds of systems. A broader background on decidability results for Petri nets and counter machines can for example be found in [10, 1].

System	strong bisimulation	weak bisimulation
Petri nets	✗	✗
Counter machines	✗	✗
One-counter Petri nets	✓	✗
One-counter machines	✓	✗
Unlabelled Petri nets	✗	✗
Lossy Petri nets	✗	✗
Lossy counter machines	✗	✗

Table 2.1: Decidability Results for Weak and Strong Bisimulation Equivalence, Partly Taken from [2]

In 1995, Jančar proved that both, strong and weak bisimilarity, is undecidable for Petri nets [3]. This result carries over to counter machines. In [4], Jančar shows that strong bisimilarity is decidable for the subclass of pushdown automata with only one stack symbol (apart from the symbol indicating the bottom of the stack). The stack is thus serving as a counter. This subclass of pushdown automata is equivalent to one-counter machines what implies that strong bisimilarity must also be decidable for one-counter nets. Weak bisimilarity, on the other hand, is undecidable for one-counter nets and therefore also for one-counter machines as proved by Mayr in [6]. Srba shows in [11] that strong bisimulation equivalence is undecidable even for unlabelled Petri nets. Since the undecidability of strong bisimilarity implies the undecidability of weak bisimilarity, we also get that weak bisimilarity is undecidable for unlabelled Petri nets. Finally, Schnoebelen proves in [9] that all equivalences in Van Glabbeek's branching time-linear time spectrum [12] (thus also strong bisimilarity) are undecidable for lossy channel systems and the even weaker lossy VASSes. This implies that strong bisimilarity must also be undecidable for lossy Petri nets and lossy counter machines. Schnoebelen's result also gives us that weak bisimilarity is undecidable in these systems.

In this thesis, we prove that weak bisimilarity is still undecidable in unlabelled one-counter nets. This strengthens some of the results above. It also supports the conjecture that weak bisimilarity is undecidable for a wide range of infinite-state systems (see more in Chapter 6).

Chapter 3

Preliminaries

In this chapter, we introduce the different systems and equivalences we use. Similar definitions can be found in several publications, e.g. [6, 11, 9]. Additionally, we state important lemmas that we base our work on.

Before we define one-counter Petri nets formally, we first give some basic definitions, e.g. of transition systems and bisimilarity.

Definition 3.1 *A* **labelled transition system** T *is a tuple* $(S, Act, \longrightarrow)$ *such that:*

- *S is a (possible infinite) set of states*
- *Act is a finite set of labels*
- *$\longrightarrow \subseteq S \times Act \times S$ is the transition relation*

For a sequence of transition steps $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots s_n$ *we also write* $s_0 \xrightarrow{a_1 \ldots a_n} s_n$.

Definition 3.2 *A transition system can model internal actions with a designated action* $\tau \in Act$. *We define the* **extended transition relation** $\Rightarrow$ *writing* $s \overset{a}{\Rightarrow} t$ *iff either* $a = \tau$ *and* $s = t$ *or* $s \xrightarrow{\tau^* a \tau^*} t$. *A step* $s \overset{a}{\Rightarrow} t$ *is called a* **weak step** *whereas a step* $s \xrightarrow{a} t$ *is referred to as a* **strong step**.

Definition 3.3 *An* **unlabelled transition system** T *is a labelled transition system with* $|Act| = 1$. *In an* **unlabelled transition system with weak steps** *we additionally have*

$\tau \in Act$ such that $|Act| = 2$. For the sake of readability, we prefer to not label strong steps in unlabelled transition systems.

Definition 3.4 *In a transition system, two states s_1 and t_1 are* **bisimilar** *if there exists a relation $\mathcal{R}$ such that:*

1. *If $(s,t) \in \mathcal{R}$ and $s \xrightarrow{a} s'$, then also $t \xrightarrow{a} t'$ and $(s',t') \in \mathcal{R}$. The same holds in the other direction for steps of t.*
2. *The initial configurations (s_1,t_1) are in $\mathcal{R}$.*

In such a case, $\mathcal{R}$ is called the **bisimulation equivalence** *and we write $s_1 \sim t_1$. The relation $\sim$ is the union of all bisimulations and itself a bisimulation.*

We obtain weak bisimulation equivalence by replacing each transition step $\xrightarrow{a}$ by a weak transition step $\overset{a}{\Rightarrow}$ in the definition of (strong) bisimulation equivalence. The following characterisation of weak bisimulation equivalence is known to be equivalent and is more convenient in proofs.

Definition 3.5 *In a transition system, two states s_1 and t_1 are* **weakly bisimilar** *if there exists a relation $\mathcal{R}$ such that:*

1. *If $(s,t) \in \mathcal{R}$ and $s \xrightarrow{a} s'$, then also $t \overset{a}{\Rightarrow} t'$ and $(s',t') \in \mathcal{R}$. The same holds in the other direction for steps of t.*
2. *The initial configurations (s_1,t_1) are in $\mathcal{R}$.*

In such a case, $\mathcal{R}$ is called the **weak bisimulation equivalence** *and we write $s_1 \approx t_1$. The relation $\approx$ is the union of all weak bisimulations and itself a weak bisimulation.*

Weak bisimulation equivalence (and with the corresponding adaptations also strong bisimulation equivalence) can be modelled with a game on the transition system. The game has two players, an attacker and a defender and is played on the transition system. Starting in a configuration $(s_1,t_1) \in S^2$, the attacker makes a first step which is either $s_1 \xrightarrow{a} s_2$ or $t_1 \xrightarrow{a} t_2$ for some $s_2,t_2 \in S$ and $a \in Act$ (note that a can also by τ). The defender has to answer with a weak step $t_1 \overset{a}{\Rightarrow} t_2$ or $s_1 \overset{a}{\Rightarrow} s_2$, respectively. Like this, the game continues in a way that the attacker can choose the process in each round.

The attacker wins if at some point, he can make a step that can not be answered by the defender. The defender wins if both processes are in a deadlock or the game continues forever. If the defender wins, it holds $s_1 \approx t_1$; if the attacker wins, it holds $s_1 \not\approx t_1$. To avoid confusion, we always assume a female attacker and a male defender when speaking about bisimulation games.

The following simple fact holds by a transitivity argument for strong and weak bisimilarity.

Fact 3.6 *Let the state of the game on a transition system T be (s,t). If the attacker can force the game into a configuration (s',t') and $s' \not\sim_T t'$ (or $s' \not\approx_T t'$, respectively), then also $s \not\sim_T t$ (or $s \not\approx_T t$, respectively). Similar, if the defender can force the game into a configuration (s',t') and $s' \sim_T t'$ (or $s' \approx_T t'$, respectively), then also $s \sim_T t$ (or $s \approx_T t$, respectively).*

In this thesis we investigate in decidability questions of different subclasses of Petri nets. In general, Petri nets allow an unbounded number of tokens in each place. The reachability graph of a net indicates all reachable configurations — also called markings. For the case where only one place may contain an arbitrary number of tokens, we follow [6] and choose a simplified notation.

Definition 3.7 *A* **one-counter Petri net** *(1CPN) is a tuple (S,X,Act,Δ) in which S is the set of control states describing the configurations of all bounded places. The special symbol X denotes the unbounded place, Act is a set of atomic actions and Δ the set of transition rules. All sets S, Act and Δ are finite. A configuration of the net is given as sX^n with $s \in S$ and $n \in \mathbb{N}$. It denotes that the system is in control state s and there are n tokens in the unbounded place. Rules in Δ are of the form $s_1X^m \xrightarrow{a} s_2X^n$ where $s_1, s_2 \in S$, $a \in Act$ and $n,m \in \mathbb{N}$. A rule describes a set of transitions with label a that go out from configurations with control state s_1 and $m+k$ tokens in place X to state s_2 and $n+k$ tokens in X where $k \in \mathbb{N}$. For a configuration sX^0 we also just write s.*

As we did for transition systems, we define a weakened version of 1CPNs where the labels are omitted.

Definition 3.8 *An* **unlabelled one-counter Petri net with weak steps** (S,X,Δ) *is a 1CPN* (S,X,Act,Δ) *where* $Act = \{a,\tau\}$. *Here, a denotes the visible transition and* τ *the internal action. For strong steps* $s_1X^m \xrightarrow{a} s_2X^n$ *we just write* $s_1X^m \to s_2X^n$.

Lastly, we introduce Minsky machines again following [6]. Minsky machines are simple register machines where the registers serve as counters that can be increased, decreased and tested for zero. Although being very simple, Minsky showed that his machines are Turing complete when introducing them in 1967 (if they have at least two counters) [7].

Definition 3.9 *A* **n-counter Minsky machine** *(nCMM) is a tuple* (Q,q_0,q_{accept},C,I) *where Q is a finite set of states,* $q_0,q_{accept} \in Q$ *are the initial and the accepting state,* $C = \{c_1,\dots,c_n\}$ *is the set of counters and I is a set of instructions where every* $i \in I$ *is of one of the two forms:*

- $(q$: $c_j \coloneqq c_j$ + 1; **goto** $p)$
- $(q$: **if** $c_j = 0$ **then goto** p **else** $c_j \coloneqq c_j$ − 1; **goto** $r)$

where $j \leq n \in \mathbb{N}$ *and* $q,q',q'' \in Q$. *A configuration in a nCMM is a* $n+1$*-tuple* $(q,v_1,\dots,v_n)$ *where* $q \in Q$ *is the current state and* $v_1,\dots,v_n \in \mathbb{N}$ *denote the counter values.*

We call an n-counter Minsky machine **deterministic** *if for every* $q \in Q$*, there is at most one instruction* $i \in I$ *where i is of the form* (q: ...). *A deterministic nCMM* **accepts** *input values* $v_1,\dots,v_n$ *if the run starting in configuration* $(q_0,v_1,\dots,v_n)$ *is finite and terminates in state* q_{accept}.

Our work is based on two decidability results on Minsky machines and one-counter Petri nets.

Fact 3.10 (Proven in [7]) *For a deterministic 2-counter Minsky machine M it is in general undecidable whether M accepts* n_1,n_2.

Fact 3.11 (Proven in [6]) *For a one-counter Petri net* $P = (S,X,Act,\Delta)$ *with* $q_0,q_0' \in S$ *and* $n \in \mathbb{N}$ *it is in general undecidable whether* $q_0X^n \approx_P q_0'X^n$.

Chapter 4

Removing Labels from Rule-Based Transition Systems

In this chapter, we present a transformation for removing transition labels from infinite transition systems that are described using a finite number of rules. The transformation we propose is designed to preserve weak bisimilarity of a pair of states in the system. We describe the transformation in detail for one-counter Petri nets. As we assume a finite number of rules, the transformation is computable and is thus suitable to be applied in reductions. Furthermore, the existence of such a transformation shows that weak bisimilarity of states in 1CPNs only depends on the branching structure of the system, not on the labels.

The biggest challenge when removing the labels from a Petri net is to ensure that if the attacker takes a transition that was formerly labelled with a, then the defender still responds with a former $a-$transition. We enforce this behaviour by introducing a unique 'testing gadget' for each action. If the defender uses a wrong transition, then the attacker can win the game by entering the gadget.

More concretely, a state s is renamed to (s,a) such that a is the action of the last strong transition a player took. From state (s,a), a player can enter the testing gadget for a. The length of a testing gadget is always even to ensure that if the attacker enters the testing gadget, then the defender has to follow, otherwise he loses. If the attacker makes an encoded a-step and the defender answers with an encoded b-step (where $a \neq b$), then the two processes are in states (s,a) and (t,b). Now, the attacker can enter the testing gadget for a. If the defender enters the testing gadget for b, then the two gadgets are not weakly bisimilar by construction and he loses. If he proceeds to some other state (r,c) instead, then the attacker enters the testing gadget for c. Since the gadgets are always of even length, one process has an even number and the other one an odd number of steps left, so the attacker wins. In contrast to strong steps, τ-steps are not a problem and can be added according to the old system. In Fig. 4.1, you can find how a rule of a 1CPN P would be translated into unlabelled rules of a 1CPN $\hat{P}$.

Formally, we define the construction as follows.

Definition 4.1 *Let us be given a labelled 1CPN $P = (S, X, Act, \Delta)$ where $Act = \{a_1, \ldots, a_n\}$. We constructively define a corresponding unlabelled 1CPN $\hat{P} = (\overline{S}, X, \overline{\Delta})$ as follows:*

$$\begin{aligned}
T(a_i) &\coloneqq \{s^1_{a_i}, \ldots, s^{2i}_{a_i}\} \\
\overline{S} &\coloneqq (S \times Act) \bigcup_{a_i \in Act} T(a_i) \\
\overline{\Delta} &\coloneqq \{(s_1, a_i)X^m \to (s_2, a_j)X^n \mid a_i, a_j \neq \tau \in Act,\ s_1 X^m \xrightarrow{a_j} s_2 X^n \in \Delta\} \\
&\quad \cup \{(s_1, a_i)X^m \xrightarrow{\tau} (s_2, a_i)X^n \mid a_i \in Act,\ s_1 X^m \xrightarrow{\tau} s_2 X^n \in \Delta\} \\
&\quad \cup \{(s, a_i) \to s^1_{a_i} \mid (s, a_i) \in S \times Act\} \\
&\quad \cup \{s^k_{a_i} \to s^{k+1}_{a_i} \mid a_i \in Act,\ 0 < k < 2i,\ s^k_{a_i}, s^{k+1}_{a_i} \in T(a_i)\}
\end{aligned}$$

The construction is designed to preserve weak bisimulation equivalence of two states in the system. Formally, we aim to prove the following theorem.

Theorem 4.2 *Let $P = (S, X, Act, \Delta)$ be a 1CPN. Then $\hat{P}$ is an unlabelled 1CPN such that for any two states $s, t \in S$ and $n, m \in \mathbb{N}$ it holds that $sX^m \approx_P tX^n$ iff for all $a \in Act$, $(s,a)X^m \approx_{\hat{P}} (t,a)X^n$.*

We prove this theorem by showing that:

1. If the defender has a winning strategy for the weak bisimulation game on P, he also has one for $\hat{P}$.

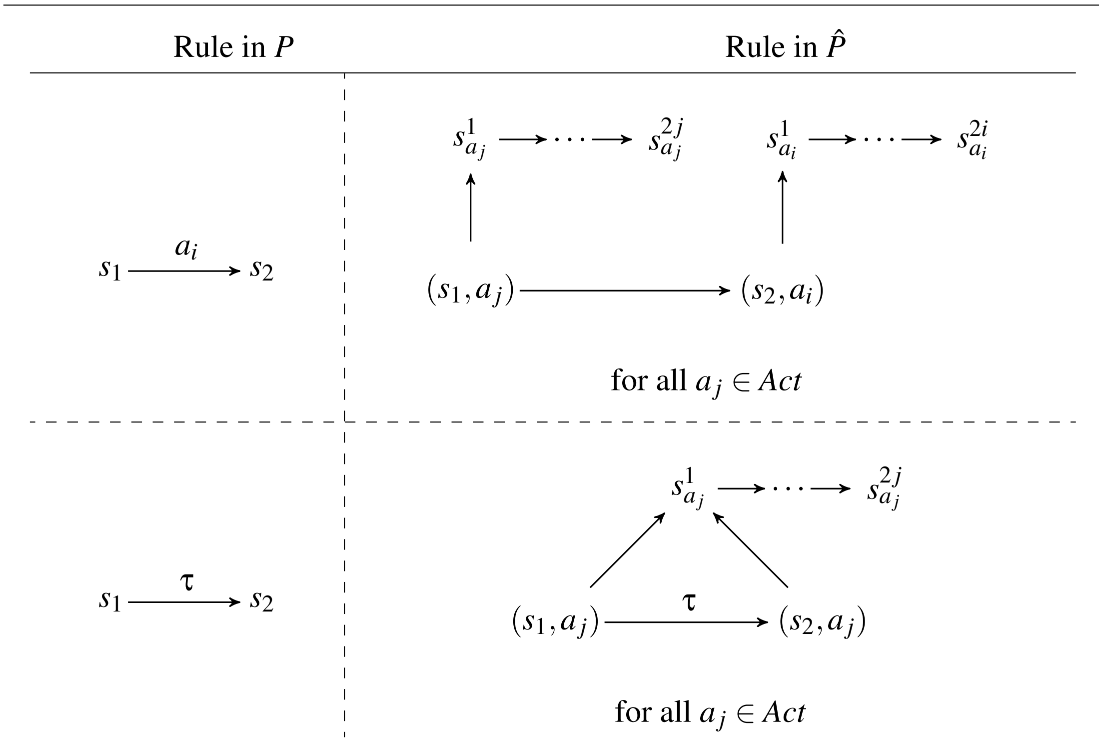

Figure 4.1: Tranformation Removing Labels from Rules in 1CPNs

2. If the attacker has a winning strategy for the weak bisimulation game on P, she also has one for $\hat{P}$.

In both cases, we describe how to translate the winning strategy from P to $\hat{P}$.

Fact 4.3 *Note that for any two $n, m \in \mathbb{N}$ and $a \in Act$, the two processes $s^1_a X^n$ and $s^1_a X^m$ are isomorphic and therefore weakly bisimilar.*

The following lemma describes an invariant that holds for every configuration of the game. As we will later see, it gives us the first direction of the proof.

Lemma 4.4 *Let $((s,a)X^m, (t,a)X^n)$ be a configuration in the weak bisimulation game on $\hat{P}$. If $sX^m \approx_P tX^n$, then one of the following statements holds:*

- *The defender can win the game on $\hat{P}$ by making the processes isomorphic in the next round.*

- *The defender can force the game on $\hat{P}$ into a configuration $((s',b)X^{m'},(t',b)X^{n'})$ where again $s'X^{m'} \approx_P t'X^{n'}$ holds.*

Proof Let us assume without loss of generality that the attacker makes a step starting in $(s,a)X^m$, the other case is similar. By construction, she can choose between three different options.

- She goes $(s,a)X^m \xrightarrow{\tau} (s',a)X^{m'}$ for some $s' \in S$ and $m' \in \mathbb{N}$. The same transition (with adapted states) is by construction also possible in P so assume the defender would have answered with a step $tX^n \overset{\tau}{\Rightarrow} t'X^{n'}$ in P. He can therefore answer with $(t,a)X^n \overset{\tau}{\Rightarrow} (t',a)X^{n'}$ in $\hat{P}$ and $s'X^{m'} \approx_P t'X^{n'}$ holds.
- She goes $(s,a)X^m \to (s',b)X^{m'}$ for some $s' \in S$, $b \in Act$ and $m' \in \mathbb{N}$. This corresponds by construction to a transition $sX^m \xrightarrow{b} s'X^{m'}$ in P which we assume is answered by the defender with $tX^n \xrightarrow{\tau^*} t_1X^{n_1} \xrightarrow{b} t_2X^{n_2} \xrightarrow{\tau^*} t'X^{n'}$ in P. Therefore, $s'X^{m'} \approx_P t'X^{n'}$ holds again. In $\hat{P}$, he can do the corresponding steps $(t,a)X^n \xrightarrow{\tau^*} (t_1,a)X^{n_1} \to (t_2,b)X^{n_2} \xrightarrow{\tau^*} (t',b)X^{n'}$.
- She goes $(s,a)X^m \to s_a^1X^m$. But then the defender can make the two processes isomorphic by answering $(t,a)X^n \to s_a^1X^n$. ■

For the other direction, we first state two simple lemmas about the correctness of our construction.

Lemma 4.5 *For all $s \in S$, $a,b \in Act$ and $n,m \in \mathbb{N}$ it holds that $(s,a)X^m \not\approx_{\hat{P}} s_b^1X^n$.*

Proof The following describes a winning strategy for the attacker. She first goes $(s,a)X^m \to s_a^1X^m$ which the defender has to answer by moving $s_b^1X^n \to s_b^2X^n$. By construction, there is now an odd number of strong steps left until deadlock in process $s_a^1X^m$ whereas in process $s_b^2X^n$, it is an even number. Therefore, $s_a^1X^m \not\approx_{\hat{P}} s_b^2X^m$. ■

Lemma 4.6 *For all $s,t \in S$, $a,b \in Act$ where $a \neq b$ and $n,m \in \mathbb{N}$ it holds that $(s,a)X^m \not\approx_{\hat{P}} (t,b)X^n$.*

Proof The following describes a winning strategy for the attacker. She first goes $(s,a)X^m \to s_a^1X^m$. The defender has two possibilities to answer.

- He goes $(t,b)X^n \Rightarrow (t',c)X^{n'}$ for some $t' \in S$, $c \in Act$ and $n' \in \mathbb{N}$. But by Lemma 4.5, $(t',c)X^{n'} \not\approx_{\hat{P}} (s^1_{a},)X^m$.
- He goes $(t,b)X^n \stackrel{\tau^*}{\Longrightarrow} (t_1,b)X^{n_1} \rightarrow s^1_b X^{n_1}$. But since $a \neq b$, one of the processes can by construction of $\hat{P}$ make strictly more strong steps then the other one. By always choosing the process with more steps until deadlock the attacker wins. ■

The next lemma will give us the other direction of Theorem 4.2. Similar to the defender's case, it states the necessary invariant to translate the attacker's winning strategy. We exploit the fact that if the attacker can win the bisimulation game, then he can do so within a finite number of rounds.

Lemma 4.7 *Let $((s,a)X^m,(t,a)X^n)$ be a configuration in the weak bisimulation game on $\hat{P}$. If the attacker can win every weak bisimulation game on P starting in (sX^m,tX^n) within l rounds, then one of the following statements holds:*

- *The attacker can force the game on $\hat{P}$ into some configuration $(yX^{m'},zX^{n'})$ such that $yX^{m'} \not\approx_{\hat{P}} zX^{n'}$.*
- *The attacker can force the game on $\hat{P}$ into some configuration $((s',b)X^{m'},(t',b)X^{n'})$ where she can win every game on P starting in $(s'X^{m'},t'X^{n'})$ within $l-1$ rounds.*

Proof Assume the attacker can win every game on P starting in (sX^m,tX^n) within l rounds. Without loss of generality, we assume that for configuration (sX^m,tX^n), the attacker's winning strategy for P stipulates a step $sX^m \xrightarrow{b} s'X^{m'}$, the other case is equivalent. Therefore, the attacker takes a step $(s,a)X^m \rightarrow (s',b)X^{m'}$ in $\hat{P}$. If there is no strong step left for the defender, then $(s,a)X^m \not\approx_{\hat{P}} (t,a)X^n$ and the first statement holds. Otherwise, the defender has several choices.

- He goes $(t,a)X^n \xrightarrow{\tau^*} (t',a)X^{n'} \rightarrow s^1_a X^{n'}$. By Lemma 4.5, $(s',b)X^{m'} \not\approx_{\hat{P}} s^1_a X^{n'}$ and the first statement holds.
- He goes $(t,a)X^n \xrightarrow{\tau^*} (t_1,a)X^{n_1} \rightarrow (t_2,c)X^{n_2} \xrightarrow{\tau^*} (t',c)X^{n'}$ where $c \neq b$. By Lemma 4.6, $(s',b)X^{m'} \not\approx_{\hat{P}} (t',c)X^{n'}$ and the first statement holds.
- He goes $(t,a)X^n \xrightarrow{\tau^*} (t_1,a)X^{n_1} \rightarrow (t_2,b)X^{n_2} \xrightarrow{\tau^*} (t',b)X^{n'}$. This corresponds to a weak step $tX^n \stackrel{b}{\Rightarrow} t'X^{n'}$ in P. By the definition of weak bisimilarity, the attacker can

win every game on P starting in $(s'X^{m'}, t'X^{n'})$ within $l-1$ rounds. Therefore, the second statement holds. ■

Having established the necessary invariants, we are able to prove the main theorem.

Proof (Theorem 4.2) For a 1CPN P, $\hat{P}$ is by construction an unlabelled 1CPN. Now, let $sX^m \approx_P tX^n$, i.e. the defender has a winning strategy for every game on P starting in configuration (sX^m, tX^n). Let furthermore $a \in Act$. The following describes a winning strategy for the defender in $\hat{P}$ starting in $((s,a)X^m, (t,a)X^n)$: By Lemma 4.4, either the defender can make the two processes isomorphic in the next round and wins or the game reaches a configuration $((s',b)X^{m'}, (t',b)X^{n'})$ where again $s'X^{m'} \approx_P t'X^{n'}$ holds. Like this, we can repeatedly apply Lemma 4.4 until either the processes become isomorphic and the defender wins or the game never terminates and he also wins.

For the other direction, let $(s,a)X^m \approx_{\hat{P}} (t,a)X^n$ for all $a \in Act$ and assume $sX^m \not\approx_P tX^n$. By definition of weak bisimilarity, this means that there is an $l \in \mathbb{N}$ such that the attacker can win every game on P starting in (sX^m, tX^n) within l rounds. We show by induction on l that for all $a \in Act$, the attacker has a winning strategy for any game on $\hat{P}$ starting in $((s,a)X^m, (t,a)X^n)$ which is a contradiction to $(s,a)X^m \approx_{\hat{P}} (t,a)X^n$.

$l = 0$ By Lemma 4.7, the attacker can force the game into a state $(yX^{m'}, zX^{n'})$ where $yX^{m'} \not\approx_{\hat{P}} zX^{n'}$ holds, so $(s,a)X^m \not\approx_{\hat{P}} (t,a)X^n$.

$l > 0$ By Lemma 4.7, one of the following cases holds: 1) The attacker can force the game into a state $(yX^{m'}, zX^{n'})$ where $yX^{m'} \not\approx_{\hat{P}} zX^{n'}$ holds, so also $(s,a)X^m \not\approx_{\hat{P}} (t,a)X^n$ or 2) The attacker can force the game into a configuration $((s',b)X^{m'}, (t',b)X^{n'})$ where she can win every game on P starting in $(s'X^{m'}, t'X^{n'})$ within $l-1$ rounds. By induction, $(s',b)X^{m'} \not\approx_{\hat{P}} (t',b)X^{n'}$, so $(s,a)X^m \not\approx_{\hat{P}} (t,a)X^n$.

■

Remark It is easy to see that the size of a $\hat{P}$ is polynomially bounded by the size of P (for more detail, see Chapter 6). Due to the finite number of rules in a 1CPN, the transformation is therefore computable and of polynomial complexity. It can thus be used in decidability or complexity reductions.

We now apply the transformation in a reduction to establish the undecidability of unlabelled one-counter nets. To do so, we first note the following simple fact. It states that the behaviour of a state $((s,a)X^n,(t,a)X^m)$ in a game on $\hat{P}$ only depends on s, not on a.

Fact 4.8 *Note that by construction of the transformation, for all $s,t \in S$, $a,b \in Act$ and $n,m \in \mathbb{N}$ it holds that $(s,a)X^n \approx_{\hat{P}} (t,a)X^m$ iff $(s,b)X^n \approx_{\hat{P}} (t,b)X^m$. This implies that $\exists\, a \in Act.\ (s,a)X^n \approx_{\hat{P}} (t,a)X^m$ iff $\forall a \in Act.\ (s,a)X^n \approx_{\hat{P}} (t,a)X^m$.*

Theorem 4.9 *There exists a fixed unlabelled 1CPN $P' = (S,X,\Delta)$ such that for two states $s_1,s_2 \in S$ and inputs $n_1,n_2 \in \mathbb{N}$, it is undecidable whether $s_1X^{n_1} \approx_{P'} s_2X^{n_2}$.*

Proof In [6] Mayr proves that there is a fixed 1CPN $P = (S,X,Act,\Delta)$ such that for two states $s_1,s_2 \in S$ and inputs $n_1,n_2 \in \mathbb{N}$, it is undecidable whether $s_1X^{n_1} \approx_P s_2X^{n_2}$. Having P, with Theorem 4.2, Fact 4.8 and choosing $P' := \hat{P}$ it follows that it must also be undecidable whether $(s_1,a)X^{n_1} \approx_{P'} (s_2,a)X^{n_2}$ for some fixed $a \in Act$. ∎

Corollary 4.10 *Weak Bisimilarity is in general undecidable for unlabelled one-counter Petri nets.*

Remark We presented a label removing transformation for one-counter Petri nets. Note that however, the transformation is not limited to 1CPNs. We did not change how the rules manipulate the counter so the construction can easily be applied to all kinds of transition systems that are described by a finite number of rules.

Chapter 5

Lossyness in One-Counter Petri Nets

Lossy semantics allow a system with counters, registers or channels to spontaneously decrease the counter value or lose messages, respectively. This behaviour is non-deterministic and cannot be controlled by the machine or a user. Lossyness therefore weakens a model.

For this thesis, we aimed to prove that weak bisimilarity for lossy one-counter Petri nets is still undecidable. As proposed by Mayr in [6], we planned to apply a similar technique like the one used by Schnoebelen in [9]. However, it turned out that lossy 1CPNs are more complicated than expected. Being bound to a limited time frame, we have not been able to complete the proof. But due to the case that the difficulties we faced were very unexpected, we present the approaches we considered and describe where the challenges lie.

5.1 Possible Definitions of Lossyness

For this project, we considered *classic lossyness*. It allows a system to lose an arbitrary amount of data at any time. There are also other versions of lossyness, e.g. *bounded lossyness* where only a bounded amount of data can be lost in one step. For a more elaborate overview over different kinds of lossyness see [5].

During our work, we considered different characterisations for classic lossyness in 1CPNs. The first one is the standard in literature (compare for example [9, 10]). It introduces a different operational semantics for one-counter nets.

Definition 5.1 (Version 1) *A* **lossy one-counter Petri net** *is a tuple* (S, X, Act, Δ) *where S is the set of control states, X denotes the unbounded place, Act is the set of actions and* Δ *the set of transition rules. All sets S, Act and* Δ *are finite. Rules in* Δ *are of the form* $s_1X^m \xrightarrow{a} s_2X^n$ *where* $s_1, s_2 \in S$, $a \in Act$ *and* $n, m \in \mathbb{N}$. *A rule describes a set of transitions with label a that go out from configurations with control state* s_1 *and* $m+k+i$ *tokens in place X to state* s_2 *and* $n+k-j$ *tokens in X where* $k, i, j \in \mathbb{N}$. *Here, i and j denote the counter values that are lost before and after the transition is taken.*

Note that this definition differs from the definition of normal 1CPNs in how the transition rules are interpreted. Giving a different semantics for a syntactically equal system means that with this definition, lossy 1CPNs are not a subclass of normal 1CPNs.

A alternative approach exploits the fact that we are considering one-counter nets modular internal actions. Hence, lossyness can be modelled by a τ-loop that decreases the counter.

Definition 5.2 (Version 2) *A* **lossy one-counter Petri net** *is a one-counter Petri net* (S, X, Act, Δ) *where for every* $s \in S$*, there is a rule* $sX \xrightarrow{\tau} s \in \Delta$.

With this definition, lossy 1CPNs are a syntactic subclass of normal 1CPNs. A proof for the undecidability of weak bisimilarity for this kind of system would carry over to 1CPNs and normal Petri nets. Furthermore, we could apply the transformation from Chapter 4 and obtain a proof that weak bisimilarity is undecidable for lossy, unary one-counter nets. These properties make the second version so appealing. Interestingly, the two definitions are not equivalent, not even up to weak bisimilarity. As an example, consider the following system.

$$sX \xrightarrow{a} p$$
$$sX \xrightarrow{\tau} s$$
$$s'X \xrightarrow{a} p$$

Configurations sX^1 and $s'X^1$ in the system above are not weakly bisimilar when considering Version 1 of the definition of lossyness: The attacker first goes $sX^1 \xrightarrow{\tau} sX^0$ to what the defender must respond with $s'X^1 \overset{\tau}{\Rightarrow} s'X^1$. Now the attacker can go $s'X^1 \xrightarrow{a} pX^0$ which cannot be met with a weak a-step by the defender. To make this system valid under Version 2, we have to add decreasing τ-loops to every state (in practice, we would implicitly assume the existence of these loops). Now, the defender can answer the attacker's first step with $s'X^1 \xrightarrow{\tau} s'X^0$ and the two configurations sX^1 and $s'X^1$ become weakly bisimilar.

That these two definitions are not equivalent up to weak bisimilarity exposes a conflict in the definition of lossyness. Version 1 is considered to be the standard in literature but it only allows losing tokens when the state of the system is changing. Thus, when considering weak bisimilarity in Petri Nets, Version 2 is more truthful to the intuitive characterisation of lossyness as the machine 'being able to lose data at any point without an observer noticing it'.

5.2 Approaches to the Undecidability Proof

Initially, we aimed to prove the following theorem.

Theorem 5.3 *Let a lossy one-counter Petri net (S, X, Act, Δ) be given. In general, it is undecidable whether two configurations sX^n and tX^m, where $s,t \in S$ and $m,n \in \mathbb{N}$, are weakly bisimilar.*

In [6], Mayr suggests that the

> *'undecidability result for weak bisimilarity of 1-counter nets carries over to the even weaker model of lossy 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 [9] by which one can ensure that whenever one player loses tokens then the other player wins, thus effectively ruling out lossy behaviour.'*

Using the technique mentioned by Mayr, Schnoebelen gives a proof that all relations between (strong) path inclusion and (strong) bisimulation equivalence are undecidable for lossy channel systems [9]. We decided to follow his suggestion and adapt the proof in [6] using Schnoebelen's technique. In the following, we first give a brief overview over Mayr's proof in order to precisely state the difficulties we encountered when trying to adapt it.

5.2.1 Undecidability Proof for Normal One-Counter Nets

To prove the undecidability of one-counter nets, Mayr encodes the execution of a two-counter Minsky machine into a one-counter Petri net. This encoding is then used in a reduction from the acceptability problem for a 2CMM to the weak bisimilarity problem for a 1CPN. Since the acceptability problem is known to be undecidable for 2CMMs (see Fact 3.10) weak bisimilarity must be undecidable for 1CPNs.

For the encoding, the two counters of the Minsky machine are compressed into the single counter of the net using Gödel's encoding. Thus, if in some configuration of the 2CMM, the counters have values n_1 and n_2, then the counter of the 1CPN has the value $2^{n_1}3^{n_2}$. This leads to the challenge of realising incrementation and decrementation of the counters in the encoding. Whenever the first (or the second) counter of the machine is increased by one, then the counter of the net must be multiplied by two (or three, respectively). Similarly, if the counter of the machine is decreased by one, then the counter of the net must be divided by two (or three). As an example of how the construction works, consider the implementation of a command $\langle q$: c_2 := c_2 + 1; **goto** $p\rangle$ as shown in Fig. 5.1. It is designed such that if both players play reasonable in a game starting in $(qX^n, q'X^n)$, then the net simulates the counter machine, i.e. the game proceeds to configuration $(pX^k, p'X^k)$ where $k = 3n$. States t^1 and t^3 are testing gadgets for which holds that $t^1X^n \approx t^1X^m$ iff $n = m$ and $t^1X^n \approx t^3X^m$ iff $n = 3m$. A state $G(s)$ allows a player to arbitrarily choose the counter value of the process with decreasing and increasing τ-loops before continuing to state s.

The following lemma formally states that the construction is correct. It is proven in [6].

Instruction in Minsky machine: (q : c_2 := c_2 + 1; **goto** p)

Corresponding rules in one-counter Petri net:

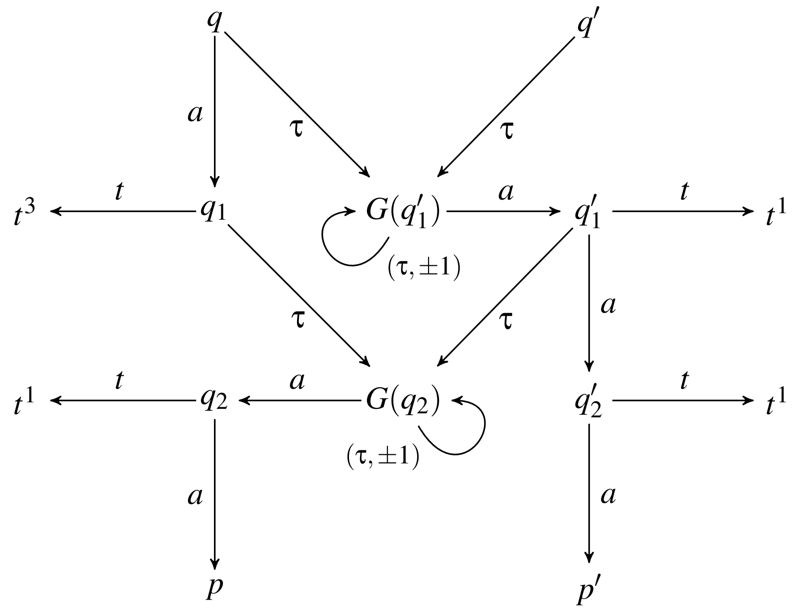

Figure 5.1: Encoding of an Instruction of a Minsky Machine into a One-Counter Net

Lemma 5.4 *Let* (q : c_i := c_2 + 1; **goto** p) *be the instruction of the Minsky 2-counter machine at control state q. Let $n_1, n_2 \in \mathbb{N}$ 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 she can either (depending on the moves of the 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 force the weak bisimulation game into the configuration $(pX^m, p'X^m)$, where $m = 2^{n_1}3^{n_2+1}$.*

To sum up the proof, the game starting in $(qX^n, q'X^n)$ is forced to proceed as follows: First, the attacker moves $qX^n \xrightarrow{a} q_1X^n$, otherwise the defender can make the two processes syntactically equal. Next, the defender goes $q'X^n \xrightarrow{a} q'_1X^k$ with $k = 3n$. If he does not choose $k = 3n$, then the attacker wins by forcing the game into the testing gadgets t^1 and t^3. Next, the attacker moves $q'_1X^k \xrightarrow{a} q'_2X^k$, otherwise the defender can make the processes syntactically equal with the help of the τ-loops in $G(q_2)$. The defender is forced to answer with $q_1X^n \overset{a}{\Rightarrow} q_2X^k$ by the thread of the testing gadget t^1. Lastly, the attacker chooses either $q_2X^k \xrightarrow{a} pX^k$ or $q'_2X^k \xrightarrow{a} p'X^k$ to which the defender responds with $q'_2X^k \overset{a}{\Rightarrow} p'X^k$ or $q_2X^k \overset{a}{\Rightarrow} pX^k$, respectively.

A similar, slightly more technical construction is presented for commands of the form $(q$: **if** $c_i = 0$ **then goto** p **else** $c_i := c_i - 1$; **goto** $r)$. To complete the undecidability proof, a final self loop is added to the accepting state q_{accept} that allows the attacker to win the game in configuration $(q_{accept}X^n, q'_{accept}X^n)$. Like this, it holds that the machine with initial state q_0 accepts values n_1, n_2 iff $q_0X^n \not\approx q'_0X^n$ where $n = 2^{n_1}3^{n_2}$.

5.2.2 Adaptation of the Proof

It is clear that the proof presented above needs some adaptations to work for lossy one-counter nets. Otherwise, the attacker could just lose tokens and the correctness lemma we stated does not hold anymore. The general idea how to punish the attacker for losing as it is presented by Schnoebelen in [9] involves that the defender is able to make the two processes syntactically equal if the attacker loses. This means that he would be able to 'cross sides', e.g. from state q' to q_1 instead of q'_1. The challenge is that the defender should only be able to cross if the attacker has lost a token. In his paper, Schnoebelen ensures this with crossing transitions that decrease the counter value (or rather drop a message for the case of channel systems). Like this, the defender would only use a crossing transition if the attacker lost a token. Otherwise, one process would have a higher counter value than the other one which gives the attacker the possibility to win the game. There are two main obstacles that hinder this idea from being realised in Mayr's construction.

First, in order to multiply the counter by two or three, the defender must be allowed to set the counter arbitrarily with a weak step, e.g. $q_1X^n \overset{a}{\Rightarrow} q_2X^k$ where $k = 3n$ (or $k = 2n$). When considering Version 2 of the definition of lossyness, this causes the following problem: In contrast to Version 1, the attacker is not forced to commit to a process or to a transition when losing tokens. Therefore, if the attacker loses, e.g. in state q'_2 with a transition $q'_2X^k \xrightarrow{\tau} q'_2X^{k-1}$, the defender needs to be able to punish the attacker immediately with a weak τ-step. This means that we need to add a rule $q_2X \xrightarrow{\tau} q'_2$. But now the defender can always win the game: If the attacker moves $q'_1X^k \xrightarrow{a} q'_2X^k$ without losing, then the defender wins by going $q_1X^n \xRightarrow{a} q'_2X^k$. This is a non-trivial problem which shows that with Version 2 of the lossyness definition, we cannot allow gadgets that permit to set a counter to an arbitrary value with a sequence of τ-steps only. That problem does not arise under the classic definition of lossyness. Since the attacker has to commit to a process and a transition when losing, we can add a rule $q_2X \xrightarrow{a} p'$, which is enough to prevent the attacker from losing on a transition $q'_2X^k \xrightarrow{a} p'X^k$.

A second problem lies in the fact that during the game, the values of the counters differ in more than a constant factor. The proof of Lemma 5.4 shows that the game proceeds in a way that at some point, the two processes are in configuration (q_1X^n, q'_1X^{3n}). This is caused by the encoding of two counters into one and is not an issue in Schnoebelen's undecidability proof for channel systems. There, the channels stay the same during the game (up to a constant factor). The different counter values cause a problem when the attacker loses a token in the process with the higher counter value. Then the defender is not able to make the two processes syntactically equal without being allowed to choose the counter value arbitrarily. This, in turn, leads to the question how to prevent the defender from always winning the game at this point.

The obstacles we encountered let us doubt that there is a straightforward solution to implement Schnoebelen's technique into Mayr's undecidability proof for one-counter nets. This is mainly due to problems that arise when encoding two counters into one. We did not expect this outcome at the beginning of the project. However, the challenges we illustrated make the question of decidability a very interesting one for future work. At the moment, we do not have a strong conjecture that either favours decidability or undecidability, for neither of the two definitions of lossyness.

Chapter 6

Evaluation

In the following, we will discuss the impact of our result in the field of process models. Furthermore, we evaluate the construction itself and compare it to the one Srba gives in [11]. Lastly, we comment on our discussion regarding lossyness in one-counter nets.

6.1 The Result

That weak bisimilarity is undecidable in unary one-counter nets is a new addition to various decidability results in the field of process models. Since unary one-counter nets are a very weak system, the result we obtain is quite strong. It carries over to all systems that subsume unlabelled 1CPNs. These results support Mayr's 'rule of thumb' he states in [6]:

> '*Weak bisimilarity is undecidable for most classes of infinite-state systems that are closed under product with finite-automata.*'

He also notes that his proof for the undecidability of weak bisimilarity in one-counter nets does not carry over to systems that are not closed under product with finite automata like basic process algebra (BPA) and basic parallel processes (BPP). As our result relies on Mayr's proof, the same obviously also holds for the proof we give in this thesis.

Mayr's proof for one-counter nets also applies to *normed 1CPNs*. In a normed 1CPN, every reachable configuration can empty its unbounded place within a finite number of steps. This is not the case for our proof. In order to norm the unlabelled 1CPN given by the construction, one would have to add a decreasing τ-loop to the end state $s_{a_i}^{2i}$ of each testing gadget to empty the counter. But then $s_{a_i}^1 X^n$ and $s_{a_i}^1 X^m$ would not be weakly bisimilar anymore which is a requirement for the correctness proof (see Fact 4.3).

6.2 The Construction

The construction itself is considerably light and straightforward. The size of the new system is polynomially bounded by the size of the old system. More concretely, the blowup is $|S| \times |Act|$ for the states that mimic the old system. For the testing gadgets, the number of states we add is

$$\sum_{a_i \in Act} 2i = |Act|^2 + |Act|.$$

As the construction is polynomially bounded, it might also be interesting to transfer complexity results to unlabelled nets.

A similar construction to remove labels from transition systems is given by Srba in [11]. Compared to our approach, he does not encode the action into the names of the states. Instead, for a transition $s \xrightarrow{a} t$, he adds a testing gadget for a between s and t. Additionally, to distinguish the testing states from original states, he adds a gadget to each original state that is longer than any of the testing gadgets. This prevents the defender from spontaneously entering the gadget. We achieve a similar effect by making the testing gadgets of even length.

Personally, we think that our construction is slightly more intuitive and easier to present. Pushing information like transition labels into the states is an easy, old trick that serves our purposes very well. However, Srba showed that his construction applies to Petri nets as well as pushdown automata and preserves strong bisimilarity and model checking of action-based modal μ-calculus formulae. This is a much broader result than what we aimed for in this thesis. Of course, this implies the question whether our construction could be used in other contexts, too.

6.3 Lossyness in One-Counter Nets

When starting the project, we hoped to also come up with a proof for the undecidability of weak bisimilarity in lossy one-counter nets or even lossy and unlabelled 1CPNs. However, we have not been able to apply Schnoebelen's technique [9] successfully, even though it seemed promising for such a proof. It seems as if the encoding of two counters into one is an existential problem for the proof. We can therefore conclude that the decidability question is still open and much more interesting than it might seem at the first sight. We also stated different definitions for lossyness in one-counter systems where one definition gives the attacker more options that the other one. Thus, it might even be the case that the decidability question is answered differently for different definitions of lossyness.

Chapter 7

Conclusion

We presented a construction to transform a labelled one-counter net into an unlabelled one preserving weak bisimilarity in the system. Using this transformation, we have been able to prove that weak bisimilarity is undecidable for unary (unlabelled) one-counter nets. The construction we described is comparable simple and very straightforward. This result contributes to the active research in the field of process models and supports the assumption that weak bisimilarity is undecidable for systems that are closed under product with finite automata.

We also discussed some obstacles we encountered when trying to prove that weak bisimilarity is also undecidable in lossy one-counter nets. It became clear that a proof for either decidability or undecidability is a interesting challenge for future work. At this point, we doubt that a reduction from two-counter Minsky machines using Schnoebelen's technique works out. However, there is a vast range of systems and undecidable problems that could be taken into consideration for a reduction. For example, one might try to encode an undecidable problem in standard one-channel systems into the weak bisimilarity problem in one-counter nets. In contrast to extended channel systems, standard ones do not allow an explicit test for zero. Encoding a standard one-channel system into a one-counter net would include the challenge of dealing with different messages in the channel.

Furthermore, it might be interesting to further explore the different definitions of lossyness in Petri nets and other systems. The discussion in Section 5.1 showed that one might apply a non-standard definition of lossyness when considering systems modulo internal actions. This alternative definition is actually not equivalent to what one normally finds in literature. In future work, one might therefore discuss which definition yields a better model for the specific system one is interested in, especially if it was found that the definitions yield different decidability results.

Bibliography

[1] Javier Esparza and Mogens Nielsen. Decidability issues for Petri nets. ***BRICS Report Series***, 1(8), 1994.

[2] Jana Hofmann. Undecidability of weak bisimulation equivalence of unary 1-counter petri nets with lossyness. Informatics Research Proposal, April 2017. University of Edinburgh.

[3] Petr Jančar. Undecidability of bisimilarity for Petri nets and some related problems. ***Theoretical Computer Science***, 148(2):281–301, 1995.

[4] Petr Jančar. Decidability of bisimilarity for one-counter processes. ***Information and Computation***, 158(1):1–17, 2000.

[5] Richard Mayr. Undecidable problems in unreliable computations. In ***Latin American Symposium on Theoretical Informatics***, pages 377–386. Springer, 2000.

[6] Richard Mayr. Undecidability of weak bisimulation equivalence for 1-counter processes. In ***International Colloquium on Automata, Languages, and Programming***, pages 570–583. Springer, 2003.

[7] Marvin L Minsky. ***Computation: finite and infinite machines***. Prentice-Hall, Inc., 1967.

[8] Adam Petri. ***Kommunikation mit Automaten***. PhD disstertation, Institut für Instrumentelle Mathematik, University of Bonn, West Germany, 1962. In German.

[9] Philippe Schnoebelen. Bisimulation and other undecidable equivalences for lossy channel systems. In ***International Symposium on Theoretical Aspects of Computer Software***, pages 385–399. Springer, 2001.

[10] Philippe Schnoebelen. Lossy counter machines decidability cheat sheet. ***RP***, 6227:51–75, 2010.

[11] Jiří Srba. On the power of labels in transition systems. In ***International Conference on Concurrency Theory***, pages 277–291. Springer, 2001.

[12] Rob J Van Glabbeek. The linear time-branching time spectrum. In ***International Conference on Concurrency Theory***, pages 278–297. Springer, 1990.