On the Power of Labels in Transition Systems

Jiří Srba*

BRICS**
Dept. of Computer Science, University of Aarhus, Denmark
srba@brics.dk

Abstract. In this paper we discuss the role of labels in transition systems with regard to bisimilarity and model checking problems. We suggest a general reduction from labelled transition systems to unlabelled ones, preserving bisimilarity and satisfiability of μ-calculus formulas. We apply the reduction to the class of transition systems generated by Petri nets and pushdown automata, and obtain several decidability/complexity corollaries for unlabelled systems. Probably the most interesting result is undecidability of strong bisimilarity for unlabelled Petri nets.

1 Introduction

Formal methods for verification of infinite-state systems have been an active area of research with a number of positive decidability results. In particular, verification techniques for concurrent systems defined by process algebras like CCS, ACP or CSP, pushdown systems, Petri nets, process rewrite systems and others have attracted a lot of attention. There are two central questions about decidability (complexity) of equivalence and model checking problems:

- **Equivalence checking** (see [Mol96]):
  Given two (infinite-state) systems, are they equal with regard to some equivalence notion?
- **Model checking** (see [BE97]):
  Given an (infinite-state) transition system and a formula ϕ of some suitable logic, does the system satisfy the property described by ϕ?

Both these problems have an interesting and unifying aspect in common. They can be defined independently on the computational model by means of *labelled transition systems*. All the models mentioned above give rise to a certain type of (infinite) labelled transition system and this is considered to be their desired semantics. Equivalence and model checking problems can be defined purely in terms of these transition systems.

In the first part of the paper we discuss the role of labels of such transition systems. There are two aspects of the branching structure described by a labelled

* The author is supported in part by the GACR, grant No. 201/00/0400.
** **B**asic **R**esearch in **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

transition system T. First, given a state of T, there can be several outgoing edges with different labels. Second, given a state of T and a label a, there can be several outgoing edges under the same label a. We claim that for our purposes only the second property is the essential one. In other words, given a labelled transition system, we can construct another transition system where all edges are labelled by the same label, i.e., the labels are in fact completely irrelevant. We call such systems *unlabelled transition systems*. What is important is the fact that our construction preserves the answers to both the questions we are interested in — equivalence checking (and we have chosen *strong bisimilarity* as the notion of equivalence) and model checking with *action-based modal μ-calculus* as the chosen logic for expressing properties of labelled transition systems.

In the second part we focus on two specific classes of infinite-state systems, namely *Petri nets* and *pushdown systems*. Petri nets are a typical example of fully parallel models of computation, whereas pushdown systems can model sequential stack-like process behaviours. Both Petri nets and pushdown systems generate (in general infinite) labelled transition systems. The question is whether the transformed unlabelled transition systems (given by the construction mentioned in the previous paragraph) are still definable by the chosen formalism of Petri nets resp. pushdown automata. The answer is shown to be positive for both our models — there are even polynomial time transformations. This implies several decidability/complexity results about bisimilarity and model checking problems for unlabelled Petri nets and pushdown systems.

Probably the most interesting corollary is the application of the transformation to Petri nets. We prove that strong bisimilarity for unlabelled Petri nets (where the set of labels is a singleton set) is undecidable. This is stronger result than undecidability of strong bisimilarity for labelled Petri nets given by Jancar [Jan95]. The undecidability for unlabelled Petri nets contrasts to a positive decidability result for the subclass of Petri nets which are deterministic [Jan95,Vog92], i.e., for any marking M and a label a there is at most one outgoing transition from M labelled by a. This again demonstrates that the role of labels is not important for decidability questions and what is crucial is the branching structure induced by transitions with the same label.

Note: full and extended version of this paper appears as [Srb01].

2 Basic definitions

Definition 1 (Labelled transition system). *A* labelled transition system *is a triple $T = (S, \mathcal{A}ct, \longrightarrow)$ where S is a set of* states *(or* processes*), $\mathcal{A}ct$ is a set of* labels *(or* actions*) such that $S \cap \mathcal{A}ct = \emptyset$, and $\longrightarrow \subseteq S \times \mathcal{A}ct \times S$ is a* transition relation*, written $\alpha \stackrel{a}{\longrightarrow} \beta$ for $(\alpha, a, \beta) \in \longrightarrow$.*

In what follows we assume that $\mathcal{A}ct$ is a finite set. As usual we extend the transition relation to the elements of $\mathcal{A}ct^*$. We also write $\alpha \longrightarrow^* \beta$ iff $\exists w \in \mathcal{A}ct^*$ such that $\alpha \stackrel{w}{\longrightarrow} \beta$. A state β is *reachable* from a state α, iff $\alpha \longrightarrow^* \beta$. Moreover, we write $\alpha \not\longrightarrow$ for $\alpha \in S$ iff there is no $\beta \in S$ and $a \in \mathcal{A}ct$ such that $\alpha \stackrel{a}{\longrightarrow} \beta$. We call a labelled transition system *normed* iff $\forall s \in S.\ \exists s' \in S$: $s \longrightarrow^* s' \not\longrightarrow$.

Definition 2. *Let* $T = (S, \mathcal{A}ct, \longrightarrow)$ *be a labelled transition system and* $s \in S$. *By* T_s *we denote a labelled transition system restricted to states of* T *reachable from* s. *More precisely,* $T_s = (S_s, \mathcal{A}ct, \longrightarrow_s)$ *where* $S_s = \{s' \in S \mid s \longrightarrow^* s'\}$ *and* $s_1 \stackrel{a}{\longrightarrow}_s s_2$ *iff* $s_1 \stackrel{a}{\longrightarrow} s_2$ *and* $s_1, s_2 \in S_s$.

Now, we introduce the notion of *(strong) bisimilarity.*

Definition 3 (Bisimulation). *Let* $T = (S, \mathcal{A}ct, \longrightarrow)$ *be a labelled transition system. A binary relation* $R \subseteq S \times S$ *is a relation of* bisimulation *iff whenever* $(\alpha, \beta) \in R$ *then for each* $a \in \mathcal{A}ct$:

- *if* $\alpha \stackrel{a}{\longrightarrow} \alpha'$ *then* $\beta \stackrel{a}{\longrightarrow} \beta'$ *for some* β' *such that* $(\alpha', \beta') \in R$
- *if* $\beta \stackrel{a}{\longrightarrow} \beta'$ *then* $\alpha \stackrel{a}{\longrightarrow} \alpha'$ *for some* α' *such that* $(\alpha', \beta') \in R$.

Two states $\alpha, \beta \in S$ *are* bisimilar in T, *written* $\alpha \sim_T \beta$, *iff there is a bisimulation* R *such that* $(\alpha, \beta) \in R$.

Bisimilarity has also an elegant characterisation in terms of *bisimulation games* [Tho93,Sti95]. A bisimulation game on a pair of states $\alpha, \beta \in S$ is a two-player game of an "attacker" and a "defender". The attacker chooses one of the states and makes an $\stackrel{a}{\longrightarrow}$-move for some $a \in \mathcal{A}ct$. The defender must respond by making an $\stackrel{a}{\longrightarrow}$-move from the other state under the same label a. Now the game repeats, starting from the new processes. If one player cannot move, the other player wins. If the game is infinite, the defender wins. States α and β are bisimilar iff the defender has a winning strategy (and non-bisimilar iff the attacker has a winning strategy).

Definition 4 (Unlabelled transition system). *Let* $T = (S, \mathcal{A}ct, \longrightarrow)$ *be a labelled transition system. We call* T unlabelled transition system *whenever* $\mathcal{A}ct$ *is a singleton set, i.e.,* $|\mathcal{A}ct| = 1$.

Remark 1. If it is the case that $|\mathcal{A}ct| = 1$ then (for our purposes) we simply write $\longrightarrow$ instead of $\stackrel{a}{\longrightarrow}$. We also forget about the second component in the definition of a labelled transition system, i.e., we can denote an unlabelled transition system by $T = (S, \longrightarrow)$ where $\longrightarrow \subseteq S \times S$.

We define a powerful logic for labelled transition systems — modal μ-calculus.

Definition 5 (Syntax of modal μ-calculus). *Let* $\mathcal{V}ar$ *be a set of variables and* $\mathcal{A}ct$ *a set of action labels such that* $\mathcal{V}ar \cap \mathcal{A}ct = \emptyset$. *The syntax of modal* μ*-calculus is defined as follows:*

$$\phi ::= \mathsf{tt} \mid X \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \langle a \rangle \phi \mid \mu X.\phi$$

where tt *stands for "true",* X *ranges over* $\mathcal{V}ar$ *and* a *over* $\mathcal{A}ct$. *There is a standard restriction on the formulas: we consider only formulas where each occurrence of a variable* X *is within a scope of an even number of negation symbols.*

Given a labelled transition system $T = (S, \mathcal{A}ct, \longrightarrow)$, we interpret a formula ϕ as follows. Assume a valuation $\mathcal{V}al : \mathcal{V}ar \to 2^S$.

$$
\begin{aligned}
[\![\mathsf{tt}]\!]_{\mathcal{V}al,T} &= S \\
[\![X]\!]_{\mathcal{V}al,T} &= \mathcal{V}al(X) \\
[\![\phi_1 \wedge \phi_2]\!]_{\mathcal{V}al,T} &= [\![\phi_1]\!]_{\mathcal{V}al,T} \cap [\![\phi_2]\!]_{\mathcal{V}al,T} \\
[\![\neg\phi]\!]_{\mathcal{V}al,T} &= S \setminus [\![\phi]\!]_{\mathcal{V}al,T} \\
[\![\langle a\rangle\phi]\!]_{\mathcal{V}al,T} &= \{s \mid \exists s'.\ (s \stackrel{a}{\longrightarrow} s' \ \wedge \ s' \in [\![\phi]\!]_{\mathcal{V}al,T})\} \\
[\![\mu X.\phi]\!]_{\mathcal{V}al,T} &= \bigcap\{S' \subseteq S \mid [\![\phi]\!]_{\mathcal{V}al[S'/X],T} \subseteq S'\}
\end{aligned}
$$

Here $\mathcal{V}al[S'/X]$ stands for a valuation function such that $\mathcal{V}al[S'/X](X) = S'$ and $\mathcal{V}al[S'/X](Y) = \mathcal{V}al(Y)$ for $X \neq Y$. We say that a formula ϕ is satisfied in a state s of T, and we write $T, s \models \phi$, if for all valuations $\mathcal{V}al$ we have $s \in [\![\phi]\!]_{\mathcal{V}al,T}$.

Remark 2. The logic defined above without the fixed-point operator $\mu X.\phi$ is called *Hennessy-Milner logic* [HM85].

3 From labelled to unlabelled transition systems

In this section we present a transformation from labelled transition systems to unlabelled ones, preserving bisimilarity and satisfiability of μ-calculus formulas.

Let $T = (S, \mathcal{A}ct, \longrightarrow)$ be a labelled transition system. We define a transformed unlabelled transition system $\widehat{T} = (\widehat{S}, \longrightarrow)$. We reuse the relation symbol $\longrightarrow$ without causing confusion, since in the system T it is a ternary relation and in $\widehat{T}$ it is a binary relation. W.l.o.g. assume that $\mathcal{A}ct = \{1, 2, \ldots, n\}$ for some $n > 0$. We define the system $\widehat{T} = (\widehat{S}, \longrightarrow)$ as follows:

$$
\begin{aligned}
\widehat{S} \quad &= S \cup \{r^k_{(s,a,s')} \mid 0 \leq k \leq a \ \wedge \ s \stackrel{a}{\longrightarrow} s'\} \cup \{d^k_s \mid s \in S \ \wedge \ 0 \leq k \leq n\} \\
\longrightarrow &= \{(s, r^0_{(s,a,s')}),\ (r^0_{(s,a,s')}, s') \mid s \stackrel{a}{\longrightarrow} s'\} \cup \\
&\quad\ \{(r^k_{(s,a,s')}, r^{k+1}_{(s,a,s')}) \mid s \stackrel{a}{\longrightarrow} s' \ \wedge \ 0 \leq k < a\} \cup \\
&\quad\ \{(s, d^0_s) \mid s \in S\} \cup \{(d^k_s, d^{k+1}_s) \mid s \in S \ \wedge \ 0 \leq k < n\}.
\end{aligned}
$$

For a better understanding of the transformation take a look at Figure 1 where a way how to transform a transition $s \stackrel{a}{\longrightarrow} s'$ is drawn. The idea consists in splitting each transition $s \stackrel{a}{\longrightarrow} s'$ labelled by $a \in \mathbb{N}_0$ with an intermediate state (the $r^0_{(s,a,s')}$ state) out of which goes a newly added linear path of length a. The d_s states add a linear path of length $n+1$ to each state from S and serve for distinguishing the r-states from the original ones.

Notice that if T is a finite-state system then the size of $\widehat{T}$ is polynomially bounded by the size of T. In fact, we could add only one linear path of length $n+1$ with appropriate links into the path starting in the states from S and in the r^0-states. However, for technical convenience in Section 4, we use the previously described construction.

Remark 3. It is an easy observation that $\widehat{T}$ is a normed transition system.

3.1 Bisimilarity

Let $T = (S, \mathcal{A}ct, \longrightarrow)$ be a labelled transition system and let $s \in S$. We define a *set of finite norms of* s by $\mathcal{N}(s) = \{|w| \mid \exists s' \in S : s \stackrel{w}{\longrightarrow} s' \not\longrightarrow\}$ where $|w|$ is the length of w. The following proposition is a standard one.

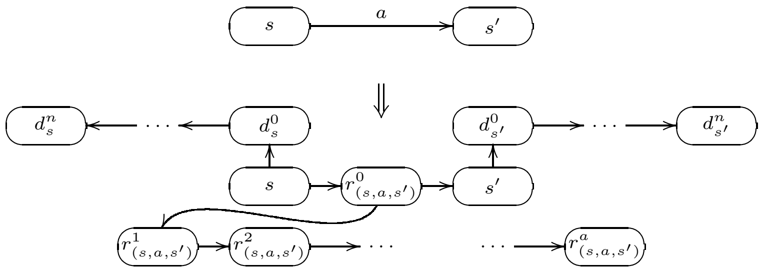

Fig. 1. Transformation of a transition $s \stackrel{a}{\longrightarrow} s'$

Proposition 1. *Let* $T = (S, \mathcal{A}ct, \longrightarrow)$ *be a labelled transition system and* $s_1, s_2 \in S$. *Then* $s_1 \sim_T s_2$ *implies that* $\mathcal{N}(s_1) = \mathcal{N}(s_2)$.

Our aim is to show that for a pair of states s_1 and s_2 of a labelled transition system T holds that $s_1 \sim_T s_2$ if and only if $s_1 \sim_{\widehat{T}} s_2$.

Lemma 1. *Let* $T = (S, \mathcal{A}ct, \longrightarrow)$ *be a labelled transition system and* $s_1, s_2 \in S$ *be a pair of states. If* $s_1 \sim_T s_2$ *then* $s_1 \sim_{\widehat{T}} s_2$.

Proof. We can naturally define a wining strategy for the defender in $\widehat{T}$ under the assumption that $s_1 \sim_T s_2$. Details can be found in [Srb01]. □

Before showing the other implication, we prove the following property.

Property 1. The attacker in $\widehat{T}$ has a winning strategy from any pair of states $s_1, s_2 \in \widehat{S}$ such that $s_1 \notin S$ and $s_2 \in S$, or $s_1 \in S$ and $s_2 \notin S$.

Proof. Assume w.l.o.g. that $s_1 \notin S$ and $s_2 \in S$. The other case is symmetric. There are three possibilities if $s_1 \notin S$.

- Let $s_1 = d_s^k$ for some $s \in S$ and $0 \le k \le n$, or $s_1 = r^k_{(s,a,s')}$ for some $s, s' \in S$, $a \in \mathcal{A}ct$ and $0 < k \le a$. In both these cases $n + 1 \notin \mathcal{N}(s_1)$ and $n + 1 \in \mathcal{N}(s_2)$. Because of Proposition 1 we get $s_1 \not\sim_{\widehat{T}} s_2$ and the attacker in $\widehat{T}$ has a winning strategy.
- Let $s_1 = r^0_{(s,a,s')}$ for some $s, s' \in S$ and $a \in \mathcal{A}ct$. Now the attacker has the following winning strategy in $\widehat{T}$. He makes a move $r^0_{(s,a,s')} \longrightarrow r^1_{(s,a,s')}$. Assume a defender's answer $s_2 \longrightarrow s_2'$ for an arbitrary $s_2' \in \widehat{S}$. Obviously either $n \in \mathcal{N}(s_2')$ or $n + 2 \in \mathcal{N}(s_2')$ and $\max\,[\mathcal{N}(r^1_{(s,a,s')})] < n$. Again, using Proposition 1, the attacker has a winning strategy. □

Lemma 2. *Let* $T = (S, \mathcal{A}ct, \longrightarrow)$ *be a labelled transition system and* $s_1, s_2 \in S$ *be a pair of states. If* $s_1 \sim_{\widehat{T}} s_2$ *then* $s_1 \sim_T s_2$.

Proof. Knowing that the defender has a winning strategy in $\widehat{T}$ from s_1 and s_2, we establish a winning strategy for the defender in T from s_1 and s_2. Suppose that the attacker's move in T is $s_i \xrightarrow{a} s'_i$ for $i \in \{1,2\}$. Then it is possible to perform a series of two moves $s_i \longrightarrow r^0_{(s_i,a,s'_i)} \longrightarrow s'_i$ in $\widehat{T}$. Because of Property 1, the defender in $\widehat{T}$ has a response to this series of moves only by performing $s_{3-i} \longrightarrow r^0_{(s_{3-i},b,s'_{3-i})} \longrightarrow s'_{3-i}$ for some $b \in \mathcal{A}ct$ and $s'_{3-i} \in S$ where

$$s'_1 \sim_{\widehat{T}} s'_2. \tag{1}$$

Notice that $a = b$, otherwise the attacker has a winning strategy in $\widehat{T}$ from $r^0_{(s_i,a,s'_i)}$ and $r^0_{(s_{3-i},b,s'_{3-i})}$ by performing a move $r^0_{(s_i,a,s'_i)} \longrightarrow r^1_{(s_i,a,s'_i)}$. Using Property 1, the defender must answer with $r^0_{(s_{3-i},b,s'_{3-i})} \longrightarrow r^1_{(s_{3-i},b,s'_{3-i})}$. However, the attacker has a winning strategy now since $a-1 \in \mathcal{N}(r^1_{(s_i,a,s'_i)})$ and $a-1 \notin \mathcal{N}(r^1_{(s_{3-i},b,s'_{3-i})})$ whenever $a \neq b$ — Proposition 1. This implies that the defender in T can perform $s_{3-i} \xrightarrow{a} s'_{3-i}$ and because of (1), the defender in T has a winning strategy from s'_1 and s'_2. Thus $s_1 \sim_T s_2$. $\square$

By Lemma 1 and Lemma 2 we can conclude with the following theorem.

Theorem 1. *Let $T = (S, \mathcal{A}ct, \longrightarrow)$ be a labelled transition system and $s_1, s_2 \in S$ be a pair of states. Let $\widehat{T}$ be the corresponding unlabelled transition system. Then*

$$s_1 \sim_T s_2 \quad \textit{if and only if} \quad s_1 \sim_{\widehat{T}} s_2.$$

3.2 Model checking

We turn our attention to the model checking problem now. We show that there is a polynomial time transformation of any μ-calculus formula ϕ into $\widehat{\phi}$ such that $T, s \models \phi$ iff $\widehat{T}, s \models \widehat{\phi}$. When interpreting a μ-calculus formula on an unlabelled transition system $\widehat{T}$, we write $\Diamond$ instead of $\langle a \rangle$, since $a \in \mathcal{A}ct$ is the only label and hence it is irrelevant. We also define a dual operator $\Box$ as $\Box\phi \equiv \neg\Diamond\neg\phi$ and ff as $\mathsf{ff} \equiv \neg\mathsf{tt}$.

Let $T = (S, \mathcal{A}ct, \longrightarrow)$ be a labelled transition system such that $\mathcal{A}ct = \{1, 2, \ldots, n\}$ and let $\widehat{T} = (\widehat{S}, \longrightarrow)$ be the corresponding unlabelled system. First of all, we write a formula $\mathcal{L}(a)$ such that

$$[\![\mathcal{L}(a)]\!]_{\mathcal{V}al',\widehat{T}} = \{r^0_{(s,a,s')} \mid \exists s, s' \in S : s \xrightarrow{a} s'\} \tag{2}$$

for any valuation $\mathcal{V}al' : \mathcal{V}ar \to 2^{\widehat{S}}$. We define $\mathcal{L}(a) \equiv \Diamond^{n+1}\mathsf{tt} \wedge \Diamond(\Box^a\mathsf{ff} \wedge \Diamond^{a-1}\mathsf{tt})$ where $\Diamond^0\phi \equiv \phi$ and $\Diamond^{k+1}\phi \equiv \Diamond(\Diamond^k\phi)$, and similarly $\Box^0\phi \equiv \phi$ and $\Box^{k+1}\phi \equiv \Box(\Box^k\phi)$. Let $\widehat{T}, s_1 \models \mathcal{L}(a)$. The left subformula in $\mathcal{L}(a)$, namely $\Diamond^{n+1}\mathsf{tt}$, ensures that the state s_1 is not of the form $r^k_{(s,b,s')}$ for $k > 0$, nor of the form d^k_s for $k \geq 0$. The second subformula in the conjunction says that there is a one step transition from s_1, reaching a state s'_1 of the form $r^1_{(s,b,s')}$ — should $s'_1 \in S$, or s'_1 be of the form $r^0_{(s,b,s')}$, or s'_1 be of the form d^0_s, then the formula $\Box^a\mathsf{ff}$ can never be satisfied. Moreover, the formula $\Box^a\mathsf{ff}$ guarantees that there are at most $a-1$

transitions from $r^1_{(s,b,s')}$ and the formula $\Diamond^{a-1}\mathtt{tt}$ finally ensures that at least $a-1$ transitions can be performed from $r^1_{(s,b,s')}$. Hence $a=b$ and (2) is established.

Let us now consider another formula defined by $\mathcal{S}tate \equiv \Diamond\mathtt{tt} \wedge \Box\Diamond^n\mathtt{tt}$. Obviously, $[\![\mathcal{S}tate]\!]_{\mathcal{V}al',\widehat{T}} = S$ for any valuation $\mathcal{V}al' : \mathcal{V}ar \to 2^{\widehat{S}}$. We are now ready to define $\widehat{\phi}$ for a given μ-calculus formula ϕ. The definition follows:

$$\begin{aligned}
\widehat{\mathtt{tt}} &= \mathtt{tt} \wedge \mathcal{S}tate\\
\widehat{X} &= X \wedge \mathcal{S}tate\\
\widehat{\phi_1 \wedge \phi_2} &= \widehat{\phi_1} \wedge \widehat{\phi_2} \wedge \mathcal{S}tate\\
\widehat{\neg\phi} &= \neg\widehat{\phi} \wedge \mathcal{S}tate\\
\widehat{\mu X.\phi} &= (\mu X.\widehat{\phi}) \wedge \mathcal{S}tate\\
\widehat{\langle a\rangle\phi} &= \Diamond\big(\mathcal{L}(a) \wedge \Diamond\widehat{\phi}\big) \wedge \mathcal{S}tate.
\end{aligned}$$

Theorem 2. *Let $T=(S,\mathcal{A}ct,\longrightarrow)$ be a labelled transition system and $s \in S$. Let ϕ be a μ-calculus formula. Then*

$$T, s \models \phi \quad \text{if and only if} \quad \widehat{T}, s \models \widehat{\phi}.$$

Proof. By structural induction on ϕ it is provable that

$$[\![\phi]\!]_{\mathcal{V}al,T} = [\![\widehat{\phi}]\!]_{\mathcal{V}al',\widehat{T}}$$

for arbitrary valuations $\mathcal{V}al : \mathcal{V}ar \to 2^S$ and $\mathcal{V}al' : \mathcal{V}ar \to 2^{\widehat{S}}$ such that $\mathcal{V}al(X) = \mathcal{V}al'(X) \cap S$ for all $X \in \mathcal{V}ar$. Full proof can be found in [Srb01]. $\Box$

Remark 4. Let us consider temporal operators $EF\phi$ and $EG\phi$ defined by $EF\phi \equiv \mu X.\phi \vee \langle -\rangle X$ and $EG\phi \equiv \neg\mu X.\neg\phi \vee \big(\neg\langle -\rangle\neg X \wedge \langle -\rangle\mathtt{tt}\big)$ such that $\langle -\rangle\phi \equiv \bigvee_{a\in\mathcal{A}ct}\langle a\rangle\phi$. We define the transformed formulas $\widehat{EF\phi}$ (using only EF operator) and $\widehat{EG\phi}$ (using only EG operator) as follows:

$$\begin{aligned}
\widehat{EF\phi} &= EF\widehat{\phi} \;\wedge\; \mathcal{S}tate\\
\widehat{EG\phi} &= EG\left(\big(\mathcal{S}tate \vee \textstyle\bigvee_{a\in\mathcal{A}ct}\mathcal{L}(a)\big) \;\wedge\; \mathcal{S}tate \implies \widehat{\phi}\right) \;\wedge\; \mathcal{S}tate.
\end{aligned}$$

Note that still $[\![\widehat{\phi}]\!]_{\mathcal{V}al',\widehat{T}} \subseteq S$ for any formula ϕ and any valuation $\mathcal{V}al' : \mathcal{V}ar \to 2^{\widehat{S}}$. Let $s \in S$. Then $T, s \models EF\phi$ iff $\widehat{T}, s \models \widehat{EF\phi}$. If moreover T_s satisfies condition

$$\forall s' \in S_s.\ \exists s'' \in S_s.\ \exists a \in \mathcal{A}ct : s' \overset{a}{\longrightarrow} s'' \qquad (3)$$

then $T, s \models EG\phi$ iff $\widehat{T}, s \models \widehat{EG\phi}$. This enables to transform formulas of even weaker logics than modal μ-calculus (such as Hennessy-Milner logic, possibly equipped with the operator EF, respectively EG) into unlabelled formulas of the same logic. Hennessy-Milner logic with the operators EF and EG is called *unified system of branching-time logic* (UB) [BAMP83] and the fragments of UB containing only the operator $EF\phi$ ($EG\phi$) are referred to as EF-logic (EG-logic).

Similarly, the until operators $E\,[\phi U\psi]$ and $A\,[\phi U\psi]$ of CTL [CE81] — defined by $E\,[\phi U\psi] \equiv \mu X.\psi \vee (\phi \wedge \langle -\rangle X)$ and $A\,[\phi U\psi] \equiv \mu X.\psi \vee \big(\phi \wedge \langle -\rangle\mathtt{tt} \wedge \neg\langle -\rangle\neg X\big)$

— can be transformed:

$$\widehat{E\ [\phi U \psi]} = E\ [(\mathcal{S}tate \implies \widehat{\phi})\ U\ \widehat{\psi}]\ \wedge\ \mathcal{S}tate$$
$$\widehat{A\ [\phi U \psi]} = \widehat{\chi^{AU}} \quad \text{where } \chi^{AU} = \neg\Big(E\ [\neg\psi\ U\ (\neg\phi \wedge \neg\psi)]\ \vee\ EG(\neg\psi)\Big).$$

In the case of $A\ [\phi U \psi]$ we use the equivalence $A\ [\phi U \psi] \iff \chi^{AU}$ from [CES86]. Again, for any $s \in S$ it holds that $T, s \models E\ [\phi U \psi]$ iff $\widehat{T}, s \models \widehat{E\ [\phi U \psi]}$. Moreover $T, s \models A\ [\phi U \psi]$ iff $\widehat{T}, s \models \widehat{A\ [\phi U \psi]}$ under the assumption of condition (3). This enables to transform also the logic CTL.

4 Applications

In this section we show how the previous results can be applied to bisimilarity/model checking of infinite-state systems. We focus in particular on a typical representative of parallel models — Petri nets (see e.g. [Pet81]) — and sequential processes — pushdown systems (see e.g. [Mol96]). We have to show that the class of transition systems generated by these models is closed under the transformation from labelled to unlabelled systems as presented in the previous section.

First of all, we remind the reader of the fact that our transformation works immediately for finite-state transition systems. In the following corollary we consider the model checking problem with these logics: Hennessy-Milner logic, EF-logic, EG-logic, UB, CTL and modal μ-calculus.

Corollary 1. *Let $T = (S, \mathcal{A}ct, \longrightarrow)$ be a finite-state labelled transition system, i.e., $|S|, |\mathcal{A}ct| < \infty$. There is a polynomial time reduction from the bisimilarity (model) checking problem for T to the bisimilarity (model) checking problem for $\widehat{T}$, where $\widehat{T}$ is an unlabelled (and finite-state) transition system.*

Proof. Immediately from Theorem 1, Theorem 2 and Remark 4. In the case of EG-logic, UB and CTL we can ensure the validity of condition (3) of Remark 4 by adding a self-loop $s \stackrel{u}{\longrightarrow} s$ (u is a fresh action) to every state $s \in S$ such that $s \not\longrightarrow$. This does not influence satisfiability of EG, UB and CTL formulas. □

4.1 Petri nets

It is a well known fact that the bisimilarity checking problem is undecidable for labelled Petri nets [Jan95]. The technique of the proof is based on a reduction from the counter machine of Minsky and the labelling is essential for the reduction. It is also known that bisimilarity is decidable for the class of Petri nets which are deterministic up to bisimilarity [Jan95], i.e., $\mathcal{F}$-deterministic nets of Vogler [Vog92]. Bisimilarity between a labelled Petri net and a finite-state system is decidable [JM95,JKM98] and EXPSPACE-hard (see e.g. comments in [May00]).

Model checking of even weak temporal logics on labelled transition systems generated by Petri nets is quite pessimistic. The only decidable logic is (trivially)

Hennessy-Milner logic. The EF-logic is undecidable [Esp97] and model checking with EG is also undecidable, even for BPP [EK95] — BPP is a strict subclass of labelled Petri nets where each transition has exactly one input place.

Definition 6 (Labelled Petri net). *A* labelled Petri net *is a tuple* $N = (P, T, F, L, \lambda)$*, where* P *is a finite set of* places*,* T *is a finite set of* transitions *such that* $T \cap P = \emptyset$*,* $F \subseteq (P \times T) \cup (T \times P)$ *is a* flow relation*,* L *is a finite set of* labels *and* $\lambda : T \to L$ *is a* labelling function.

A *marking* M of a net N is a mapping $M : P \to \mathbb{N}_0$, i.e., each place is assigned a nonnegative number of *tokens*. We define ${}^\bullet t = \{p \mid (p,t) \in F\}$ and $t^\bullet = \{p \mid (t,p) \in F\}$ for a transition $t \in T$. We say that $t \in T$ is *enabled* in a marking M iff $\forall p \in {}^\bullet t.\ M(p) > 0$. If t is enabled in M then it can be *fired*, producing a marking M' such that:

- $M'(p) = M(p)$ for all $p \in \big(P \smallsetminus ({}^\bullet t \cup t^\bullet)\big) \cup ({}^\bullet t \cap t^\bullet)$
- $M'(p) = M(p) - 1$ for all $p \in {}^\bullet t \smallsetminus t^\bullet$
- $M'(p) = M(p) + 1$ for all $p \in t^\bullet \smallsetminus {}^\bullet t$.

Then we write $M[t\rangle M'$. W.l.o.g. we assume that if $M[t_1\rangle M'$ and $M[t_2\rangle M'$, then $\lambda(t_1) \neq \lambda(t_2)$ for any pair of markings M, M' and transitions t_1, t_2.

Definition 7 (Labelled transition system $T(N)$).
Let $N = (P, T, F, L, \lambda)$ *be a labelled Petri net. We define a corresponding labelled transition system* $T(N)$ *as* $T(N) = ([P \to \mathbb{N}_0], L, \longrightarrow)$ *where* $M \xrightarrow{a} M'$ *whenever* $M[t\rangle M'$ *and* $a = \lambda(t)$ *for* $M, M' \in [P \to \mathbb{N}_0]$ *and* $t \in T$.

Now, we define unlabelled Petri nets.

Definition 8 (Unlabelled Petri net). *An* unlabelled Petri net *is a labelled Petri net* $N = (P, T, F, L, \lambda)$ *such that* $|L| = 1$.

Remark 5. Whenever $|L| = 1$, let us say $L = \{a\}$, we omit L and λ from the definition of the net N and instead of $M \xrightarrow{a} M'$ in $T(N)$ we simply write $M \longrightarrow M'$.

Let $N = (P, T, F, L, \lambda)$ be a labelled Petri net. W.l.o.g. assume that $L = \{1, \ldots, n\}$ for some $n > 0$. We construct an unlabelled Petri net $N' = (P', T', F')$ and a mapping $\psi : (P \to \mathbb{N}_0) \to (P' \to \mathbb{N}_0)$ such that $\widehat{T(N)}_{M_1}$ and $T(N')_{\psi(M_1)}$ are isomorphic unlabelled transition systems for any marking M_1 of N. Let us recall that $\widehat{T(N)}_{M_1}$ is the transition system restricted to markings reachable from M_1 and $T(N')_{\psi(M_1)}$ is restricted to markings reachable from $\psi(M_1)$ — see Definition 2. The net N' is defined as follows:

$$P' = P \cup \{p_t^k \mid t \in T \ \wedge\ 0 \le k \le \lambda(t)\} \cup \{p_c\} \cup \{d^k \mid 0 \le k \le n\}$$

$$T' = \{t^{in}, t^{out} \mid t \in T\} \cup \{l_t^k \mid t \in T \ \wedge\ 0 \le k < \lambda(t)\} \cup \{l^k \mid 0 \le k \le n\}$$

$$\begin{aligned} F' = {} & \{(p, t^{in}) \mid (p,t) \in F\} \cup \{(t^{out}, p) \mid (t,p) \in F\} \cup \\ & \{(t^{in}, p_t^0), (p_t^0, t^{out}) \mid t \in T\} \cup \\ & \{(p_t^k, l_t^k), (l_t^k, p_t^{k+1}) \mid t \in T \ \wedge\ 0 \le k < \lambda(t)\} \cup \\ & \{(p_c, t^{in}), (t^{out}, p_c) \mid t \in T\} \cup \\ & \{(p_c, l^0)\} \cup \{(l^k, d^k), (d^k, l^{k+1}) \mid 0 \le k < n\} \cup \{(l^n, d^n)\}. \end{aligned}$$

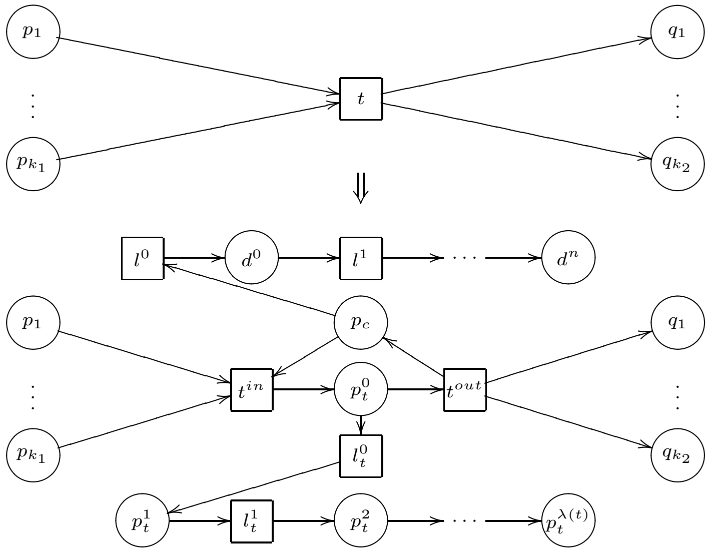

Fig. 2. Transformation of a transition t

In this construction each transition t with input places $p_1, \ldots, p_{k_1}$ and output places $q_1, \ldots, q_{k_2}$ is transformed into a set of transitions shown in Figure 2. Now, we give the mapping ψ. Let $M \in (P \to \mathbb{N}_0)$. Then $\psi(M) : P' \to \mathbb{N}_0$ is defined by

$$\psi(M)(p) = \begin{cases} 1 & \text{if } p = p_c \\ M(p) & \text{if } p \in P \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 3. *Let $N = (P, T, F, L, \lambda)$ be a labelled Petri net and $N' = (P', T', F')$ the unlabelled Petri net defined above. Then $\widehat{T(N)}_{M_1}$ and $T(N')_{\psi(M_1)}$ are isomorphic unlabelled transition systems for any $M_1 \in [P \to \mathbb{N}_0]$.*

Proof. Assume that $\widehat{T(N)}_{M_1} = (S_1, \longrightarrow_1)$ and $T(N')_{\psi(M_1)} = (S_2, \longrightarrow_2)$. Recall that $S_1 \subseteq [P \to \mathbb{N}_0] \cup \{r^k_{(M,\lambda(t),M')} \mid M[t\rangle M' \;\wedge\; 0 \le k \le \lambda(t)\} \cup \{d^k_M \mid M \in [P \to \mathbb{N}_0] \;\wedge\; 0 \le k \le n\}$ and $S_2 \subseteq [P' \to \mathbb{N}_0]$. We define a mapping $f : S_1 \to S_2$ by

$$f(s_1) = \begin{cases} \psi(s_1) & \text{if } s_1 \in [P \to \mathbb{N}_0] \\ \overline{M} & \text{if } s_1 = r^k_{(M,\lambda(t),M')} \text{ such that } M[t\rangle M' \\ \overline{\overline{M}} & \text{if } s_1 = d^k_M \text{ such that } M \in [P \to \mathbb{N}_0] \end{cases}$$

where

$$\overline{M}(p) = \begin{cases} M(p) & \text{if } p \in P \smallsetminus {}^\bullet t \\ M(p) - 1 & \text{if } p \in {}^\bullet t \\ 1 & \text{if } p = p_t^k \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \overline{\overline{M}}(p) = \begin{cases} M(p) & \text{if } p \in P \\ 1 & \text{if } p = d^k \\ 0 & \text{otherwise.} \end{cases}$$

Let $s_1 \longrightarrow_1 s_1'$ for some $s_1, s_1' \in S_1$. It can be easily seen that $f(s_1) \longrightarrow_2 f(s_1')$. On the other hand, let $M_2 \longrightarrow_2 M_2'$ and $M_2 = f(s_1)$ for some $s_1 \in S_1$ and $M_2, M_2' \in S_2$. Then there exists $s_1' \in S_1$ such that $M_2' = f(s_1')$ and $s_1 \longrightarrow_1 s_1'$. This implies that f is surjective and moreover f is trivially injective. Hence, $\widehat{T(N)}_{M_1}$ and $T(N')_{\psi(M_1)}$ are isomorphic unlabelled transition systems. □

Theorem 3. *Let N be a labelled Petri net, and M_1, M_2 a pair of markings in N and ϕ a μ-calculus formula. There is a polynomial time reduction producing an unlabelled and normed Petri net N', a pair of markings $\psi(M_1), \psi(M_2)$ in N' and a μ-calculus formula $\widehat{\phi}$ such that*

$$M_1 \sim_{T(N)} M_2 \quad \textit{if and only if} \quad \psi(M_1) \sim_{T(N')} \psi(M_2)$$

and

$$T(N), M_1 \models \phi \quad \textit{if and only if} \quad T(N'), \psi(M_1) \models \widehat{\phi}.$$

Proof. By Lemma 3 and Theorems 1 and 2. Normedness is by Remark 3. □

Since the bisimilarity checking problem and model checking problems with *EF*-logic and *EG*-logic are undecidable [Jan95,Esp97,EK95] for labelled Petri nets, we obtain the following undecidability results for unlabelled and normed Petri nets. In the case of model checking problems we use Remark 4 and the fact that undecidability of model checking with *EG*-logic can be proved by standard "weak" simulation of a 2-counter machine and we can easily ensure the validity of condition (3) for the Petri net simulating the 2-counter machine.

Corollary 2. *Bisimilarity checking problem for unlabelled and normed Petri nets is undecidable.*

Corollary 3. *Model checking problems with EF-logic and EG-logic for unlabelled and normed Petri nets are undecidable.*

Since the bisimilarity checking problem between a labelled Petri net and a finite-state system is EXPSPACE-hard (see comments e.g. in [May00]), we get also the following corollary.

Corollary 4. *Bisimilarity checking problem between an unlabelled and normed Petri net and a finite-state system is EXPSPACE-hard.*

4.2 Pushdown systems

It is known that the bisimilarity checking problem for pushdown processes is decidable [Sén98] and PSPACE-hard [May00]. PSPACE-hard is also the bisimilarity checking problem between a pushdown process and a finite-state system [May00] — this problem is moreover in EXPTIME [JKM98].

Model checking pushdown processes with modal μ-calculus is decidable and EXPTIME-complete [Wal96]. This means that the model checking problem with EF-logic, EG-logic and CTL is also in EXPTIME. The model checking problems with these logics are PSPACE-hard — see e.g. [May98]. Moreover, model checking with EF-logic and CTL is known ([Wal00]) to be PSPACE-complete and EXPTIME-complete, respectively. The exact complexity of model checking with EG-logic is unknown, however, it seems to be EXPTIME-complete by modification of arguments from [Wal00].

Definition 9 (Pushdown system). *A* pushdown system Δ *is a tuple* $\Delta = (Q, \Gamma, \mathcal{A}ct, \longrightarrow_\Delta)$ *where* Q *is a finite set of* control states*,* Γ *is a finite* stack alphabet *such that* $Q \cap \Gamma = \emptyset$*,* $\mathcal{A}ct$ *is a finite* input alphabet*, and* $\longrightarrow_\Delta \subseteq Q \times \Gamma \times \mathcal{A}ct \times Q \times \Gamma^*$ *is a finite* ($|\longrightarrow_\Delta| < \infty$) transition relation*, written* $pA \stackrel{a}{\longrightarrow}_\Delta q\alpha$ *for* $(p, A, a, q, \alpha) \in \longrightarrow_\Delta$*.*

Definition 10 (Labelled transition system $T(\Delta)$).
Let $\Delta = (Q, \Gamma, \mathcal{A}ct, \longrightarrow_\Delta)$ *be a pushdown system. We define a corresponding labelled transition system* $T(\Delta)$ *as* $T(\Delta) = (S, \mathcal{A}ct, \longrightarrow)$ *where* $S = \{p\beta \mid p \in Q \ \wedge\ \beta \in \Gamma^*\}$ *and* $p\beta \stackrel{a}{\longrightarrow} q\gamma$ *iff* $\beta = A\beta'$*,* $\gamma = \alpha\beta'$ *and* $pA \stackrel{a}{\longrightarrow}_\Delta q\alpha$*.*

Our aim is to transform Δ into an unlabelled pushdown system such that bisimilarity and model checking are preserved. For technical convenience, we assume from now on that Γ contains a distinct "dummy" symbol Z such that $pZ \not\longrightarrow$ for any $p \in Q$. Then trivially

$$p_1\beta_1 \sim_{T(\Delta)} p_2\beta_2 \quad \text{if and only if} \quad p_1\beta_1 Z \sim_{T(\Delta)} p_2\beta_2 Z \tag{4}$$

$$T(\Delta), p_1\beta_1 \models \phi \quad \text{if and only if} \quad T(\Delta), p_1\beta_1 Z \models \phi \tag{5}$$

for any $p_1, p_2 \in Q$, $\beta_1, \beta_2 \in \Gamma^*$ and a μ-calculus formula ϕ. In particular, all reachable states from $p\beta Z$ are of the form $q\beta' Z$ where $p, q \in Q$ and $\beta, \beta' \in \Gamma^*$.

Definition 11 (Unlabelled pushdown system). *An* unlabelled pushdown system *is a pushdown system* $\Delta = (Q, \Gamma, \mathcal{A}ct, \longrightarrow_\Delta)$ *such that* $|\mathcal{A}ct| = 1$*.*

Remark 6. Whenever $|\mathcal{A}ct| = 1$, let us say $\mathcal{A}ct = \{a\}$, we omit $\mathcal{A}ct$ from the definition of the pushdown system Δ and instead of $pA \stackrel{a}{\longrightarrow}_\Delta q\alpha$ we simply write $pA \longrightarrow_{\Delta'} q\alpha$ where $\Delta' = (Q, \Gamma, \longrightarrow_{\Delta'})$ and $\longrightarrow_{\Delta'} \subseteq Q \times \Gamma \times Q \times \Gamma^*$.

Let $\Delta = (Q, \Gamma, \mathcal{A}ct, \longrightarrow_\Delta)$ be a pushdown system such that $Z \in \Gamma$ is the "dummy" stack symbol. W.l.o.g. assume that $\mathcal{A}ct = \{1, \ldots, n\}$ for some $n > 0$. We construct an unlabelled pushdown system $\Delta' = (Q, \Gamma', \longrightarrow_{\Delta'})$ where $\Gamma \subseteq \Gamma'$

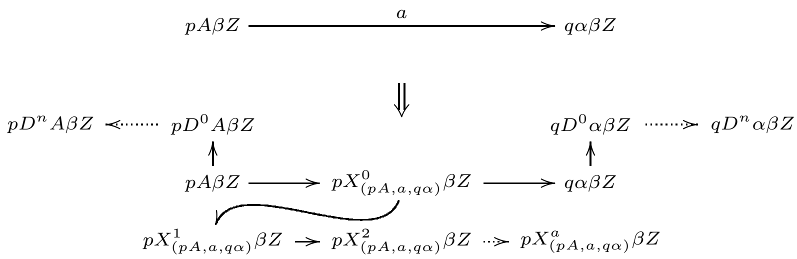

Fig. 3. Transformation of a transition $pA\beta Z \xrightarrow{a} q\alpha\beta Z$

such that $\widehat{T(\Delta)_{p_1\alpha_1 Z}}$ and $T(\Delta')_{p_1\alpha_1 Z}$ are isomorphic unlabelled transition systems for any $p_1 \in Q$ and $\alpha_1 \in \Gamma^*$. Again, see Definition 2 for the notation of transition systems restricted to reachable states from $p_1\alpha_1 Z$. The system Δ' is defined as follows:

$$\begin{aligned}
\Gamma' \quad &= \Gamma \cup \{X^k_{(pA,a,q\alpha)} \mid pA \xrightarrow{a}_\Delta q\alpha \ \wedge\ 0 \le k \le a\} \cup \{D^k \mid 0 \le k \le n\} \\
\longrightarrow_{\Delta'} &= \{(p, A, p, X^0_{(pA,a,q\alpha)}),\ (p, X^0_{(pA,a,q\alpha)}, q, \alpha) \mid pA \xrightarrow{a}_\Delta q\alpha\} \cup \\
&\quad \{(p, X^k_{(pA,a,q\alpha)}, p, X^{k+1}_{(pA,a,q\alpha)}) \mid pA \xrightarrow{a}_\Delta q\alpha \ \wedge\ 0 \le k < a\} \cup \\
&\quad \{(p, A, p, D^0A) \mid p \in Q \ \wedge\ A \in \Gamma\} \cup \\
&\quad \{(p, D^k, p, D^{k+1}) \mid p \in Q \ \wedge\ 0 \le k < n\}.
\end{aligned}$$

Notice that in particular $pX^a_{(pA,a,q\alpha)}\beta Z \not\longrightarrow$ and $pD^n\beta Z \not\longrightarrow$ for any $\beta \in \Gamma'^*$. Graphical representation showing the transformation of $pA\beta Z \xrightarrow{a} q\alpha\beta Z$ where $\beta \in \Gamma^*$ and $pA \xrightarrow{a}_\Delta q\alpha$ can be seen in Figure 3.

Lemma 4. *Let $\Delta = (Q, \Gamma, \mathcal{A}ct, \longrightarrow_\Delta)$ be a pushdown system containing $Z \in \Gamma$. Let $\Delta' = (Q, \Gamma', \longrightarrow_{\Delta'})$ be the unlabelled pushdown system defined above. Then $\widehat{T(\Delta)_{p_1\alpha_1 Z}}$ and $T(\Delta')_{p_1\alpha_1 Z}$ are isomorphic unlabelled transition systems for any $p_1 \in Q$ and $\alpha_1 \in \Gamma^*$.*

Proof. Immediately from the construction. Notice that it is important that any reachable state in $T(\Delta')_{p_1\alpha_1 Z}$ ends with Z. In particular, from any state of the form $p\beta Z$ where $p \in Q$ and $\beta \in \Gamma^*$ (even if $\beta = \epsilon$) the following transition is possible in $T(\Delta')$: $p\beta Z \longrightarrow pD^0\beta Z$. □

Theorem 4. *Let Δ be a pushdown system, and $p_1\beta_1, p_2\beta_2$ a pair of states in $T(\Delta)$ and ϕ a μ-calculus formula. There is a polynomial time reduction producing an unlabelled and normed pushdown system Δ', a pair of states $\psi(p_1\beta_1), \psi(p_2\beta_2)$ in $T(\Delta')$ and a μ-calculus formula $\widehat{\phi}$ such that*

$$p_1\beta_1 \sim_{T(\Delta)} p_2\beta_2 \quad \textit{if and only if} \quad \psi(p_1\beta_1) \sim_{T(\Delta')} \psi(p_2\beta_2)$$

and

$$T(\Delta), p_1\beta_1 \models \phi \quad \textit{if and only if} \quad T(\Delta'), \psi(p_1\beta_1) \models \widehat{\phi}.$$

Proof. Directly from Lemma 4 together with (4) and (5) — producing the mapping ψ such that $\psi(p\beta) = p\beta Z$ for $p \in Q$ and $\beta \in \Gamma^*$ — and from Theorems 1 and 2. Normedness is because of Remark 3. □

Since the bisimilarity checking problem between a pushdown system and a finite-state system is PSPACE-hard [May00] (this is trivially also a lower bound for two pushdown systems), and because the model checking problems with CTL and Hennessy-Milner logic are EXPTIME-complete resp. PSPACE-complete [Wal00,May98], we obtain the following corollaries. In the case of CTL we use Remark 4 and the fact that we can easily ensure the validity of condition (3) similarly as in the proof of Corollary 1.

Corollary 5. *Bisimilarity checking problem between an unlabelled and normed pushdown system and a finite-state system (or another unlabelled and normed pushdown system) is PSPACE-hard.*

Corollary 6. *Model checking problems with CTL and Hennessy-Milner logic for unlabelled and normed pushdown systems are EXPTIME-complete and PSPACE-complete, respectively.*

The bisimilarity checking problem between a pushdown system and a finite-state system is in EXPTIME [JKM98] and PSPACE-hard [May00]. In order to establish its containment in e.g. PSPACE, it is enough to show it for unlabelled and normed pushdown systems.

Acknowledgements: I would like to thank Mogens Nielsen for his kind supervision and Daniel Polansky for his comments and suggestions.

References

[BAMP83] M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. *Acta Informatica*, 20(3):207–226, 1983.

[BE97] O. Burkart and J. Esparza. More infinite results. *Bulletin of the European Association for Theoretical Computer Science*, 62:138–159, June 1997. Columns: Concurrency.

[CE81] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logic of Programs Workshop*, volume 131 of *LNCS*, pages 52–71. Springer-Verlag, 1981.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[EK95] J. Esparza and A. Kiehn. On the model checking problem for branching time logics and basic parallel processes. In *International Conference on Computer-Aided Verification (CAV'95)*, volume 939 of *LNCS*, pages 353–366, 1995.

[Esp97] J. Esparza. Decidability of model-checking for infinite-state concurrent systems. *Acta Informatica*, 34:85–107, 1997.

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the Association for Computing Machinery*, 32(1):137–161, 1985.

[Jan95] P. Jancar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148(2):281–301, 1995.

[JKM98] P. Jancar, A. Kucera, and R. Mayr. Deciding bisimulation-like equivalences with finite-state processes. In *Proceedings of the Annual International Colloquium on Automata, Languages and Programming (ICALP'98)*, volume 1443 of *LNCS*. Springer-Verlag, 1998.

[JM95] P. Jancar and F. Moller. Checking regular properties of Petri nets. In *Proceedings of CONCUR'95*, volume 962 of *LNCS*, pages 348–362. Springer-Verlag, 1995.

[May98] R. Mayr. Strict lower bounds for model checking BPA. In *Proceedings of the MFCS'98 Workshop on Concurrency*, volume 18 of *ENTCS*. Springer-Verlag, 1998.

[May00] R. Mayr. On the complexity of bisimulation problems for pushdown automata. In *IFIP International Conference on Theoretical Computer Science (IFIP TCS'2000)*, volume 1872 of *LNCS*. Springer-Verlag, 2000.

[Mol96] F. Moller. Infinite results. In *Proceedings of CONCUR'96*, volume 1119 of *LNCS*, pages 195–216. Springer-Verlag, 1996.

[Pet81] J.L. Peterson. *Petri Net Theory and the Modelling of Systems.* Prentice-Hall, 1981.

[Sén98] G. Sénizergues. Decidability of bisimulation equivalence for equational graphs of finite out-degree. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science(FOCS-98)*, pages 120–129. IEEE Computer Society, 1998.

[Srb01] J. Srba. On the power of labels in transition systems. Technical Report RS-01-19, BRICS Research Series, 2001.

[Sti95] C. Stirling. Local model checking games. In *Proceedings of the 6th International Conference on Concurrency Theory (CONCUR'95)*, volume 962 of *LNCS*, pages 1–11. Springer-Verlag, 1995.

[Tho93] W. Thomas. On the Ehrenfeucht-Fraïssé game in theoretical computer science (extended abstract). In *Proceedings of the 4th International Joint Conference CAAP/FASE, Theory and Practice of Software Development (TAPSOFT'93)*, volume 668 of *LNCS*, pages 559–568. Springer-Verlag, 1993.

[Vog92] W. Vogler. *Modular construction and partial order semantics of Petri nets*, volume 625 of *LNCS*. Springer-Verlag, 1992.

[Wal96] I. Walukiewicz. Pushdown processes: Games and model checking. In *International Conference on Computer-Aided Verification (CAV'96)*, volume 1102 of *LNCS*, pages 62–74, 1996. To appear in Information and Computation.

[Wal00] I. Walukiewicz. Model checking CTL properties of pushdown systems. In *Proceedings Foundations of Software Technology and Theoretical Computer Science (FSTTCS'00)*, volume 1974 of *LNCS*, pages 127–138. Springer-Verlag, 2000.