Mathematical Structures in Computer Science

http://journals.cambridge.org/MSC

Complexity of weak bisimilarity and regularity for BPA and BPP

JIRÍ SRBA

Mathematical Structures in Computer Science / Volume 13 / Issue 04 / August 2003, pp 567 - 587
DOI: 10.1017/S0960129503003992, Published online: 31 July 2003

Link to this article: http://journals.cambridge.org/abstract_S0960129503003992

How to cite this article:
JIRÍ SRBA (2003). Complexity of weak bisimilarity and regularity for BPA and BPP. Mathematical Structures in Computer Science, 13, pp 567-587 doi:10.1017/S0960129503003992

Math. Struct. in Comp. Science (2003), *vol.* 13, *pp.* 567–587.
DOI: 10.1017/S0960129503003992 Printed in the United Kingdom

Complexity of weak bisimilarity and regularity for BPA and BPP

JIŘÍ SRBA[†]

BRICS[‡]
Department of Computer Science, University of Aarhus, Ny Munkegade bld. 540, DK-8000 Aarhus C, Denmark
Email: srba@brics.dk

Received 21 November 2000; revised 2 July 2001

It is an open problem whether weak bisimilarity is decidable for Basic Process Algebra (BPA) and Basic Parallel Processes (BPP). A *PSPACE* lower bound for BPA and *NP* lower bound for BPP were demonstrated by Stribrna. Mayr recently achieved a result, saying that weak bisimilarity for BPP is Π_2^P-hard. We improve this lower bound to *PSPACE*, and, moreover, prove this result for the restricted class of *normed* BPP. It is also not known whether weak regularity (finiteness) of BPA and BPP is decidable. In the case of BPP there is a Π_2^P-hardness result by Mayr, which we improve to *PSPACE*. No lower bound has previously been established for BPA. We demonstrate *DP*-hardness, which, in particular, implies both *NP* and *co-NP*-hardness. In each of the bisimulation/regularity problems we also consider the classes of normed processes. Finally, we show how the technique for proving *co-NP* lower bound for weak bisimilarity of BPA can be applied to strong bisimilarity of BPP.

1. Introduction

This paper compares the classes of purely sequential processes BPA and purely parallel processes BPP with respect to the complexity of weak bisimilarity/regularity checking. An intensive study of a variety of process algebras based on the interleaving model of CCS (Milner 1989) has taken place in recent years. Much activity has been focused on the analysis of infinite state systems. Two central questions are decidability and complexity of certain behavioural equivalences (see Moller (1996) for a survey) and verification of system properties expressed in suitable logics – model checking (see Burkart and Esparza (1997) for a survey).

In this paper we address the first question with a special focus on bisimulation equivalence. *Strong bisimulation equivalence* is known to be decidable for the classes of Basic Process Algebra (BPA) (Christensen *et al.* 1995) and Basic Parallel Processes (BPP) (Christensen *et al.* 1993). The only known lower bound is *co-NP*-hardness for unnormed BPP achieved by Mayr (Mayr 2000a). No elementary upper bound has been established for this problem. For unnormed BPA, the strong bisimilarity problem is known

[†] The author is supported in part by the GACR, grant No. 201/03/1161.
[‡] Basic Research in Computer Science, Centre of the Danish National Research Foundation.

to be in 2-*EXPTIME* (Burkart *et al.* 1995) and no lower bound is known. *Strong regularity* (*finiteness*) for BPA and BPP is also decidable (Burkart *et al.* 1996; Jancar and Esparza 1996). If we restrict ourself to *normed* processes, there are even polynomial time algorithms for strong bisimilarity/regularity of BPA and BPP (Hirshfeld *et al.* 1996a; Hirshfeld *et al.* 1996b; Kucera 1996).

However, we consider the notion of *weak bisimilarity*, which is a more general equivalence than strong bisimilarity, in the sense that it allows us to abstract from internal behaviour of processes by introducing a *silent action* τ, which is not observable (Milner 1989).

Decidability of weak bisimulation equivalence and weak regularity (finiteness) for BPA and BPP are well-known open problems. Weak bisimilarity is known to be semi-decidable for BPP (Esparza 1995) and there are also partial results, for example, by Hirshfeld (Hirshfeld 1996), showing decidability of weak bisimilarity for restricted classes of so called *totally normed* BPA and BPP. Stribrna proved *NP*-hardness for these restricted classes in Stribrna (1998b). Nevertheless, non-bisimilarity in the classes of totally normed BPA and BPP is finitely approximable (Stribrna 1998a). There is a very recent result by Stirling (Stirling 2001) showing that weak bisimilarity is decidable for a subclass of normed BPP where non-bisimilarity is not finitely approximable. Unfortunately, the result does not imply decidability for the whole class of normed BPP, which is still an open problem.

Some results are also known about weak bisimilarity of BPA/BPP with finite state systems (Jancar *et al.* 1998; Kucera and Mayr 1999). Despite the fact that weak bisimilarity and regularity are not known to be decidable, only a few lower bounds have been found so far. This could indicate that these problems might still be decidable, but probably with worse complexity and more sophisticated algorithms than for strong bisimilarity and regularity.

For weak bisimilarity in the BPA class, *PSPACE*-hardness was proved by Stribrna (Stribrna 1998b) using a reduction from the *totality problem for finite non-deterministic automata*. No lower bound had previously been established for weak regularity in this class.

In the class of BPP, weak bisimilarity appeared to be *NP*-hard (Stribrna 1998b). This result was recently improved by Mayr (Mayr 2000a) to Π_2^P (in the polynomial hierarchy). In the same paper, Π_2^P-hardness for weak regularity is proved.

There are also some results about lower bounds for strong bisimilarity of PDA (Mayr 2000b) and for model checking problems of BPA (Mayr 1998) and PDA (Walukiewicz 1996; Bouajjani *et al.* 1997).

Our contribution

We show *PSPACE*-hardness of weak bisimilarity for BPP, thus improving the Π_2^P-hardness result by Mayr, and, moreover, we prove our result for the restricted class of *normed* BPP. This result can be transformed to weak regularity for BPP, thus achieving a *PSPACE* lower bound (again even for normed processes).

For the class of BPA we prove *DP*-hardness of weak regularity, which, in particular, means both *NP* and *co-NP*-hardness. Moreover, *NP*-hardness can be transformed to the normed case.

All these results hold also for PA (Process Algebra (Baeten and Weijland 1990)), which is a natural 'union' of BPA and BPP in which we are allowed to use both sequential and parallel composition.

In the final section we prove a *co-NP* lower bound for strong bisimilarity of BPP. This result was recently demonstrated by Mayr (Mayr 2000a) in Theorem 4. We give a substantially different and, hopefully, simpler proof using a similar technique to that used for weak bisimilarity of BPA. This is our justification for including the result here.

The paper is structured as follows. In Section 2 we give definitions of labelled transition systems, process rewrite systems, weak bisimilarity and the corresponding bisimilarity game. In Section 3 we show *PSPACE*-hardness of weak bisimilarity and regularity for BPP, and in Section 4 we demonstrate that weak regularity for BPA is both *NP* and *co-NP*-hard. Section 5 deals with strong bisimilarity of BPP, and includes the proof that the problem is *co-NP* hard using techniques introduced in Section 4. Finally, in Section 6 we conclude by summarising the current state of knowledge of weak bisimilarity and regularity problems for BPA and BPP.

2. Basic definitions

Transition systems (Plotkin 1981; Moller 1996) are widely used to give semantics to concurrent processes. Processes are understood as nodes of a certain transition system and the transition relation is defined in a compositional way.

Definition 2.1 (Labelled transition system). A *labelled transition system* is a triple $(S, \mathscr{A}ct, \longrightarrow)$ where:

— S is a set of *states* (or *processes*)
— $\mathscr{A}ct$ is a set of *labels* (or *actions*)
— $\longrightarrow \subseteq S \times \mathscr{A}ct \times S$ is a *transition relation*, written $\alpha \xrightarrow{a} \beta$, for $(\alpha, a, \beta) \in \longrightarrow$.

We can use process algebras to describe an infinite transition system in a finite way. Let $\mathscr{A}ct$ and $\mathscr{C}onst$ be countable sets of *actions* and *process constants* such that $\mathscr{A}ct \cap \mathscr{C}onst = \emptyset$. Moreover, suppose that $\mathscr{A}ct$ contains a distinguishable *silent action* τ. Let $Op \subseteq \{.\,, \|\}$. We define the class of *process expressions over* Op as

$$E_{Op}^{\mathscr{C}onst} ::= \epsilon \mid X \mid E \otimes E$$

where ϵ is the *empty process*, X ranges over $\mathscr{C}onst$ and $\otimes$ ranges over Op. The operator '.' is a *sequential composition*, and '$\|$' stands for a *parallel composition*. In what follows we will not distinguish between process expressions related by a *structural congruence*, which is the smallest congruence over process expressions such that the following laws hold:

— '.' is associative
— '$\|$' is associative and commutative
— 'ϵ' is a unit for '.' and '$\|$'.

In this paper we consider the class of PA (Process Algebra (Baeten and Weijland 1990)) expressions $E^{\mathscr{C}onst}_{\{.,\|\}}$ and its natural subclasses; BPA (Basic Process Algebra, also known as context-free processes) expressions $E^{\mathscr{C}onst}_{\{.\}}$ with only sequential composition; and BPP (Basic Parallel Processes) expressions $E^{\mathscr{C}onst}_{\{\|\}}$ with only parallel composition.

Definition 2.2 (Process rewrite system). A *PA (respectively, BPA or BPP) process rewrite system* (PRS) (Mayr 2000c) is a finite set Δ of *rules* of the form $X \stackrel{a}{\longrightarrow} E$, where $X \in \mathscr{C}onst$, $a \in \mathscr{A}ct$ and $E \in E^{\mathscr{C}onst}_{\{.,\|\}}$ (respectively, $E \in E^{\mathscr{C}onst}_{\{.\}}$ or $E \in E^{\mathscr{C}onst}_{\{\|\}}$).

Let us denote the set of actions and process constants that appear in Δ as $\mathscr{A}ct(\Delta)$ and $\mathscr{C}onst(\Delta)$, respectively (note that these sets are finite). A process rewrite system Δ determines a *transition system* $(S, \mathscr{A}ct, \longrightarrow)$ where the states are process expressions over $\mathscr{C}onst(\Delta)$, that is, $S = E^{\mathscr{C}onst(\Delta)}_{Op}$, and $\mathscr{A}ct = \mathscr{A}ct(\Delta)$ is the set of labels. The *transition relation* $\longrightarrow$ is the least relation satisfying the following SOS rules (recall that '$\|$' is commutative):

$$\frac{(X \stackrel{a}{\longrightarrow} E) \in \Delta}{X \stackrel{a}{\longrightarrow} E} \qquad \frac{E \stackrel{a}{\longrightarrow} E'}{E.F \stackrel{a}{\longrightarrow} E'.F} \qquad \frac{E \stackrel{a}{\longrightarrow} E'}{E\|F \stackrel{a}{\longrightarrow} E'\|F}$$

As usual, we extend the transition relation to the elements of $\mathscr{A}ct^*$. We also write $E \longrightarrow^* E'$ whenever $E \stackrel{w}{\longrightarrow} E'$ for some $w \in \mathscr{A}ct^*$. A state E' is *reachable from a state* E iff $E \longrightarrow^* E'$.

Definition 2.3 (Weak transition relation). A *weak transition relation* $\Longrightarrow$ is defined as follows:

$$\stackrel{a}{\Longrightarrow} \stackrel{\text{def}}{=} \begin{cases} \stackrel{\tau^*}{\longrightarrow} \circ \stackrel{a}{\longrightarrow} \circ \stackrel{\tau^*}{\longrightarrow} & \text{if } a \neq \tau \\ \stackrel{\tau^*}{\longrightarrow} & \text{if } a = \tau. \end{cases}$$

We define a *process* as a pair (P, Δ), where P is a process expression and Δ is a process rewrite system. *States* of (P, Δ) are the states of the corresponding transition system. We say that a state E is *reachable* iff $P \longrightarrow^* E$. Whenever (P, Δ) has only finitely many reachable states, we call it a *finite-state process.*

Example 2.1. Consider BPA processes (X, Δ_1), (X', Δ_1) and BPP processes (Y, Δ_2), (Y', Δ_2) where Δ_1 is given by

$$\begin{array}{llll} X \stackrel{\tau}{\longrightarrow} X.A & X \stackrel{\tau}{\longrightarrow} \epsilon & X' \stackrel{\tau}{\longrightarrow} X & X' \stackrel{a}{\longrightarrow} C \\ A \stackrel{a}{\longrightarrow} \epsilon & C \stackrel{a}{\longrightarrow} C & & \end{array}$$

and Δ_2 is given by

$$\begin{array}{llll} Y \stackrel{\tau}{\longrightarrow} Y\|A & Y \stackrel{\tau}{\longrightarrow} \epsilon & Y' \stackrel{\tau}{\longrightarrow} Y & Y' \stackrel{a}{\longrightarrow} C \\ A \stackrel{a}{\longrightarrow} \epsilon & C \stackrel{a}{\longrightarrow} C. & & \end{array}$$

Fragments of transition systems generated by (X, Δ_1), (X', Δ_1), (Y, Δ_2) and (Y', Δ_2) can be seen in Figure 1. Observe that $X \stackrel{a}{\Longrightarrow} A^{\cdot(n)}$ for any $n \geqslant 0$ where $A^{\cdot(n)}$ stands for $\underbrace{A.\ldots.A}_{n\times}$.

(X, Δ_1) :

$$\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "\tau"] & X.A \arrow[r, "\tau"] \arrow[d, "\tau"] & X.A.A \arrow[r, "\tau"] \arrow[d, "\tau"] & \cdots \\
\epsilon & A \arrow[l, "a"'] & A.A \arrow[l, "a"'] & \cdots \arrow[l, "a"']
\end{tikzcd}$$

(X', Δ_1) :

$$\begin{tikzcd}
X' \arrow[r, "\tau"] \arrow[d, "a"] & X \arrow[r, "\tau"] \arrow[d, "\tau"] & X.A \arrow[r, "\tau"] \arrow[d, "\tau"] & X.A.A \arrow[r, "\tau"] \arrow[d, "\tau"] & \cdots \\
C \arrow[loop below, "a"] & \epsilon & A \arrow[l, "a"'] & A.A \arrow[l, "a"'] & \cdots \arrow[l, "a"']
\end{tikzcd}$$

(Y, Δ_2) :

$$\begin{tikzcd}
Y \arrow[r, shift left, "\tau"] \arrow[d, "\tau"] & Y\|A \arrow[l, shift left, "a"] \arrow[r, shift left, "\tau"] \arrow[d, "\tau"] & Y\|A\|A \arrow[l, shift left, "a"] \arrow[r, shift left, "\tau"] \arrow[d, "\tau"] & \cdots \arrow[l, shift left, "a"] \\
\epsilon & A \arrow[l, "a"'] & A\|A \arrow[l, "a"'] & \cdots \arrow[l, "a"']
\end{tikzcd}$$

(Y', Δ_2) :

$$\begin{tikzcd}
Y' \arrow[r, "\tau"] \arrow[d, "a"] & Y \arrow[r, shift left, "\tau"] \arrow[d, "\tau"] & Y\|A \arrow[l, shift left, "a"] \arrow[r, shift left, "\tau"] \arrow[d, "\tau"] & Y\|A\|A \arrow[l, shift left, "a"] \arrow[r, shift left, "\tau"] \arrow[d, "\tau"] & \cdots \arrow[l, shift left, "a"] \\
C \arrow[loop below, "a"] & \epsilon & A \arrow[l, "a"'] & A\|A \arrow[l, "a"'] & \cdots \arrow[l, "a"']
\end{tikzcd}$$

Fig. 1. Processes (X, Δ_1), (X', Δ_1), (Y, Δ_2) and (Y', Δ_2).

Similarly, any state of the form $A^{\|(n)}$ for $n \geqslant 0$ is reachable in (Y, Δ_2). Here $A^{\|(n)}$ stands for $\underbrace{A\|\cdots\|A}_{n\times}$. Thus we have examples of processes with infinite branching.

Important subclasses of process algebras can be obtained by an extra restriction on the involved processes – *normedness.*

Definition 2.4 (Normed processes). A process expression E is *normed* iff there is $w \in \mathscr{A}ct^*$ such that $E \xrightarrow{w} \epsilon$. A process (P, Δ) is *normed* if any state reachable from P is normed; in our case a sufficient condition for (P, Δ) to be normed is that all process constants $X \in \mathscr{C}onst(\Delta)$ are normed. We say that (P, Δ) is *totally normed* iff it is normed and, moreover, there is no transition $X \stackrel{\tau}{\Longrightarrow} \epsilon$ for any $X \in \mathscr{C}onst(\Delta)$.

We remind the reader of the fact that normedness is easily decidable in polynomial time.

Example 2.2. The process expressions $A^{\cdot(n)}$ and $A^{\|(n)}$ from Example 2.1 are normed, since $A^{\cdot(n)} \xrightarrow{a^n} \epsilon$ and $A^{\|(n)} \xrightarrow{a^n} \epsilon$. However, the processes (X', Δ_1) and (Y', Δ_2) are not normed, since the unnormed process constant C is reachable in both of them. Processes (X, Δ_1) and (Y, Δ_2) are normed but not totally normed.

Now, we introduce the concept of *weak bisimilarity* (Park 1981; Milner 1989).

Definition 2.5 (Weak bisimulation). A binary relation $R \subseteq E_{Op}^{\mathscr{C}onst} \times E_{Op}^{\mathscr{C}onst}$ over process expressions is a relation of *weak bisimulation* iff whenever $(E, F) \in R$, then for each $a \in \mathscr{A}ct$:

— if $E \overset{a}{\Longrightarrow} E'$, then $F \overset{a}{\Longrightarrow} F'$ for some F' such that $(E', F') \in R$;
— if $F \overset{a}{\Longrightarrow} F'$, then $E \overset{a}{\Longrightarrow} E'$ for some E' such that $(E', F') \in R$.

Processes (P_1, Δ_1) and (P_2, Δ_2) are *weakly bisimilar*, written $(P_1, \Delta_1) \approx (P_2, \Delta_2)$, iff there is a weak bisimulation R such that $(P_1, P_2) \in R$. Note that without loss of generality we can suppose that $\Delta_1 = \Delta_2$ since we can always consider a disjoint union of Δ_1 and Δ_2 as a new Δ.

Example 2.3. Processes $(A^{\cdot(n)}, \Delta_1)$ and $(A^{\|(n)}, \Delta_2)$ from Example 2.1 are weakly bisimilar whereas $(X, \Delta_1) \not\approx (X', \Delta_1)$ and $(Y, \Delta_2) \not\approx (Y', \Delta_2)$. For details see Example 2.4.

If we assume that τ does not appear in a process rewrite system Δ, the relations $\Longrightarrow$ and $\longrightarrow$ coincide, and we call the corresponding version of bisimilarity *strong bisimilarity* and denote it by $\sim$.

Bisimulation equivalence has an elegant characterisation in terms of *bisimulation games* (Thomas 1993; Stirling 1995).

Definition 2.6 (Bisimulation game). A bisimulation game on a pair of processes (P_1, Δ) and (P_2, Δ) is a two-player game of an 'attacker' and a 'defender'. The game is played in rounds. In each round:

— the attacker chooses one of the processes and makes an $\overset{a}{\Longrightarrow}$-move for some $a \in \mathscr{A}ct(\Delta)$; and
— the defender must respond by making an $\overset{a}{\Longrightarrow}$-move in the other process under the same action 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.

The following theorem is standard.

Theorem 2.1. The processes (P_1, Δ) and (P_2, Δ) are weakly bisimilar iff the defender has a winning strategy (and non-bisimilar iff the attacker has a winning strategy).

Remark 2.1. Note that the previous theorem is also valid in the case when we change the rules of the game slightly: the attacker is allowed to perform only a single $\overset{a}{\longrightarrow}$-move whereas the defender can respond with an $\overset{a}{\Longrightarrow}$-move. The attacker can now simulate an $\overset{a}{\Longrightarrow}$-move by a corresponding sequence of single moves.

Example 2.4. Consider Example 2.1 again. We show that $(X, \Delta_1) \not\approx (X', \Delta_1)$. The attacker's winning strategy is the following. He plays in the process X' by performing an $\xrightarrow{a}$ move and reaching the state C. The defender can only respond by $X \stackrel{a}{\Longrightarrow} A^{\cdot(n)}$ for some n. However, now the attacker can perform $n+1$ moves $C \xrightarrow{a} C$ and the defender loses since he can perform only n of $\stackrel{a}{\Longrightarrow}$-moves from the process $A^{\cdot(n)}$. Similarly $(Y, \Delta_2) \not\approx (Y', \Delta_2)$ – in this case the attacker, in addition, has to switch twice between the processes (Y, Δ_2) and (Y', Δ_2).

Let us now assume a bisimulation game with fixed number of rounds k. If the attacker cannot win during at most k rounds then the defender wins. For a pair of processes (P_1, Δ) and (P_2, Δ) we say that they are *weakly bisimilar up to level* k if the defender has a winning strategy for k rounds. We write $(P_1, \Delta) \approx_k (P_2, \Delta)$ if it is the case, and we call $\approx_k$ *weak bisimulation approximants* (Baeten *et al.* 1987; Milner 1989).

Observe that in Example 2.1, $(X, \Delta_1) \approx_k (X', \Delta_1)$ and $(Y, \Delta_2) \approx_k (Y', \Delta_2)$ for any natural number k. The defender can always generate enough process constants A using the τ actions to protect himself in k rounds for any fixed k. On the other hand $(X, \Delta_1) \not\approx (X', \Delta_1)$ and $(Y, \Delta_2) \not\approx (Y', \Delta_2)$. This is an example of BPA and BPP processes where non-bisimilarity is not finitely approximable. There are several papers (Baeten *et al.* 1987; Stribrna 1998a; Stribrna 1999; Stirling 2001) studying the bisimulation approximants and it is known that strong non-bisimilarity in the class of BPA and BPP is finitely approximable. This holds even for any pair of processes where one of them is finitely branching (Baeten *et al.* 1987). The classes of totally normed BPA and BPP with respect to weak bisimilarity are also finitely approximable (Stribrna 1998a). The only positive decidability result for a process algebra where weak non-bisimilarity is not finitely approximable is due to Stirling (Stirling 2001).

3. Hardness of weak bisimilarity and regularity for BPP

Problem:	Weak bisimilarity of (normed) BPP
Instance:	Two (normed) BPP processes (P_1, Δ) and (P_2, Δ).
Question:	$(P_1, \Delta) \approx (P_2, \Delta)$?

We show that weak bisimilarity of normed BPP is *PSPACE*-hard. We prove it by reduction from QSAT[†], which is known to be *PSPACE*-complete (Papadimitriou 1994).

Problem:	QSAT
Instance:	A natural number n and a Boolean formula ϕ in conjunctive normal form with Boolean variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$.
Question:	Is $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots \forall x_n \exists y_n . \phi$ true?

[†] This problem is also known as QBF – *Quantified Boolean formula.*

A *literal* is a variable or the negation of a variable. Let

$$C \equiv \forall x_1 \exists y_1 \forall x_2 \exists y_2 \dots \forall x_n \exists y_n . C_1 \wedge C_2 \wedge \dots \wedge C_k$$

be an instance of QSAT, where each *clause* C_j, $1 \leqslant j \leqslant k$, is a disjunction of literals. We define the following BPP processes (P_1, Δ) and (P_2, Δ), where

$$\mathscr{C}onst(\Delta) = \{Q_1, \dots, Q_k, X_1, \dots, X_n, Y_1, \dots, Y_n\}$$

and

$$\mathscr{A}ct(\Delta) = \{q_1, \dots, q_k, x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n, y\}.$$

For each i, $1 \leqslant i \leqslant n$, let

- α_i be a parallel composition of process constants from $\{Q_1, \dots, Q_k\}$ such that Q_j appears in α_i iff the literal x_i occurs in C_j (that is, if x_i is set to true, then C_j is satisfied),
- $\overline{\alpha_i}$ be a parallel composition of process constants from $\{Q_1, \dots, Q_k\}$ such that Q_j appears in $\overline{\alpha_i}$ iff the literal $\neg x_i$ occurs in C_j (that is, if x_i is set to false, then C_j is satisfied),
- β_i be a parallel composition of process constants from $\{Q_1, \dots, Q_k\}$ such that Q_j appears in β_i iff the literal y_i occurs in C_j,
- $\overline{\beta_i}$ be a parallel composition of process constants from $\{Q_1, \dots, Q_k\}$ such that Q_j appears in $\overline{\beta_i}$ iff the literal $\neg y_i$ occurs in C_j.

The set of transition rules Δ is given by:

$$
\begin{array}{lll}
X_i \xrightarrow{x_i} Y_i \| \alpha_i & X_i \xrightarrow{\overline{x}_i} Y_i \| \overline{\alpha_i} & \text{for } 1 \leqslant i \leqslant n \\[2ex]
Y_i \xrightarrow{y} X_{i+1} \| \beta_i & Y_i \xrightarrow{y} X_{i+1} \| \overline{\beta_i} & \text{for } 1 \leqslant i \leqslant n-1 \\[2ex]
Y_n \xrightarrow{y} \beta_n & Y_n \xrightarrow{y} \overline{\beta_n} & \\[2ex]
X_i \xrightarrow{q_j} X_i & Y_i \xrightarrow{q_j} Y_i & \text{for } 1 \leqslant i \leqslant n \text{ and } 1 \leqslant j \leqslant k \\[2ex]
Q_j \xrightarrow{q_j} Q_j & Q_j \xrightarrow{\tau} \epsilon & \text{for } 1 \leqslant j \leqslant k.
\end{array}
$$

Finally, let

$$P_1 \stackrel{\text{def}}{=} X_1 \| Q_1 \| Q_2 \| \dots \| Q_k \qquad \text{and} \qquad P_2 \stackrel{\text{def}}{=} X_1.$$

We can see the processes P_1 and P_2 using Petri net notation in Figure 2. This figure is only illustrative, and some transitions, namely $X_i \xrightarrow{q_j} X_i$ and $Y_i \xrightarrow{q_j} Y_i$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k$, are missing. The curly lines stand for the corresponding sets of arrows for α_i, $\overline{\alpha_i}$, β_i and $\overline{\beta_i}$, respectively. The intuition is that the attacker will be forced to play only in the process P_1, and if C is true, the defender will have the possibility to add all the process constants $\{Q_1, \dots, Q_k\}$.

Let γ be a parallel composition of elements from $\mathscr{C}onst(\Delta)$. We define the set of process constants that occur in γ as $\mathsf{set}(\gamma) \stackrel{\text{def}}{=} \{X \in \mathscr{C}onst(\Delta) \mid X \text{ occurs in } \gamma\}$, and we also define $\mathsf{set}_Q(\gamma) \stackrel{\text{def}}{=} \mathsf{set}(\gamma) \cap \{Q_1, \dots, Q_k\}$. The following proposition is an immediate consequence of the definition of Δ.

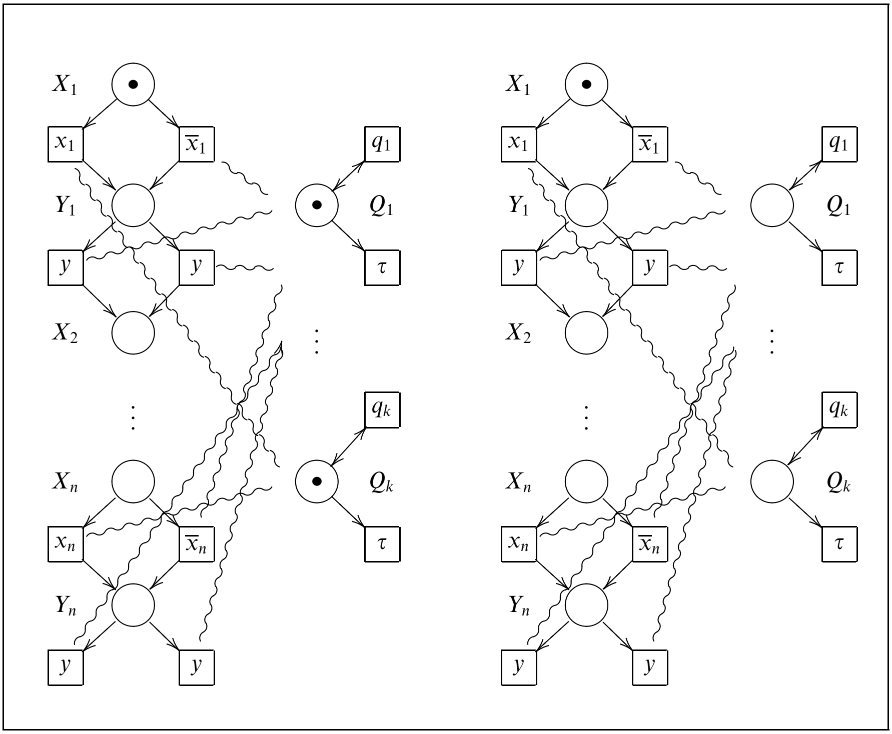

Fig. 2. The processes (P_1, Δ) and (P_2, Δ) as Petri nets.

Proposition 3.1. Let γ and γ' be parallel compositions of some process constants from $\{Q_1, \ldots, Q_k\}$. Then $\mathsf{set}_Q(\gamma) = \mathsf{set}_Q(\gamma')$ if and only if $(\gamma, \Delta) \approx (\gamma', \Delta)$.

We want to show that C is true if and only if $(P_1, \Delta) \approx (P_2, \Delta)$.

Lemma 3.1. If $(P_1, \Delta) \approx (P_2, \Delta)$, then C is true.

Proof. We show that $(P_1, \Delta) \not\approx (P_2, \Delta)$, supposing that C is false. If C is false, then $C' \stackrel{\text{def}}{=} \exists x_1 \forall y_1 \exists x_2 \forall y_2 \ldots \exists x_n \forall y_n . \neg(C_1 \wedge C_2 \wedge \ldots \wedge C_k)$ is true, and from this we claim that the attacker has a winning strategy in the bisimulation game for (P_1, Δ) and (P_2, Δ). The attacker plays only in the process P_1 (without using τ actions) performing the following sequence of actions

$$\widetilde{x}_1, y, \widetilde{x}_2, y, \ldots, \widetilde{x}_n, y$$

where $\widetilde{x}_i$, $1 \leqslant i \leqslant n$, corresponds to either x_i or $\overline{x}_i$, depending on the truth values for which the formula C' is true. It does not matter how the choice of the rule for the action y is solved. The defender can only respond by performing the same actions $\widetilde{x}_1, y, \widetilde{x}_2, y, \ldots, \widetilde{x}_n, y$ (eventually using some τ actions). The actions $\widetilde{x}_1, \ldots, \widetilde{x}_n$ are forced. For the action y there are always two possibilities, corresponding to assigning a truth value for some y_i, $1 \leqslant i \leqslant n$. Finally, the processes P_1 and P_2 are in states P_1' and P_2', respectively, such that $\mathsf{set}(P_1') = \{Q_1, \ldots, Q_k\}$ and $\mathsf{set}(P_2') \subseteq \{Q_1, \ldots, Q_k\}$. Since we assume that C' is true, there must be a clause C_j, $1 \leqslant j \leqslant k$ that is not satisfied. Hence $Q_j \notin \mathsf{set}(P_2')$, and P_2' cannot perform q_j. However, q_j is enabled in P_1' and thus the attacker has a winning strategy. This implies that $(P_1, \Delta) \not\approx (P_2, \Delta)$. □

Note that the winning strategy for the attacker in the proof above does not require any switching of sides (the attacker plays only on the side of the process P_1). For the proof of the opposite direction, let us first observe the following property of (P_1, Δ) and (P_2, Δ) above. Let δ be some state such that $\mathsf{set}(\delta) \cap \{Q_1, \dots, Q_k\} = \emptyset$, and let γ and γ' be parallel compositions of some process constants from $\{Q_1, \dots, Q_k\}$ satisfying the condition that $\mathsf{set}_Q(\gamma) \supseteq \mathsf{set}_Q(\gamma')$. Let us consider the processes $\delta \| \gamma$ and $\delta \| \gamma'$. Whenever the attacker chooses any move in the second one, the defender has an answer, which makes these two processes weakly bisimilar (using τ actions to eliminate the extra process constants Q_j from the first process, and then by Proposition 3.1). We are now ready to prove the following lemma.

Lemma 3.2. If C is true, then $(P_1, \Delta) \approx (P_2, \Delta)$.

Proof. Let P_1' and P_2' denote successors of P_1 and P_2, respectively, in the bisimulation game. The defender's strategy is to satisfy the following conditions during the game:

— $\mathsf{set}_Q(P_1') \supseteq \mathsf{set}_Q(P_2')$; and
— never delete (using τ actions) any process constant Q_j, $1 \leqslant j \leqslant k$, in the process P_2' unless it is necessary for satisfying the first condition.

Of course these conditions are true at the beginning of the game. Using the argument above this lemma, we can see that whenever the attacker makes a move in the process P_2', he immediately loses, since the defender can make the resulting processes weakly bisimilar. This means that the only possible winning strategy for the attacker is to keep playing in P_1'. However, now the defender can always fulfil the conditions of his strategy. On a move containing x_i or $\overline{x}_i$, respectively, there is only one possible response for the defender. Whenever the attacker makes a y move, the defender chooses one of the rules $Y_i \xrightarrow{y} X_{i+1} \| \beta_i$ and $Y_i \xrightarrow{y} X_{i+1} \| \overline{\beta_i}$ such that the formula $\forall x_{i+1} \exists y_{i+1} \dots \forall x_n \exists y_n . C_1 \wedge \dots \wedge C_k$ is still true. Since we have the rules $X_i \xrightarrow{q_j} X_i$ and $Y_i \xrightarrow{q_j} Y_i$ for any i, j such that $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant k$, the only possibility for the attacker to win is to perform some sequence

$$\widetilde{x}_1, y, \widetilde{x}_2, y, \dots, \widetilde{x}_n, y,$$

which may also include some τ actions, and then reach some state P_1' where $\mathsf{set}(P_1') \subseteq \{Q_1, \dots, Q_k\}$. Since C is true, the defender can always get to a corresponding state P_2' where $\mathsf{set}(P_1') = \mathsf{set}(P_2')$. Hence (using Proposition 3.1), the attacker loses again. This means that the defender has a winning strategy and so $(P_1, \Delta) \approx (P_2, \Delta)$. □

Theorem 3.1. Weak bisimilarity of normed BPP is *PSPACE*-hard.

Proof. Observe that all the process constants in Δ are normed and that the reduction is in polynomial time. The theorem is then an immediate consequence of Lemma 3.1 and Lemma 3.2. □

Corollary 3.1. Weak bisimilarity of BPP is *PSPACE*-hard.

Proof. The statement follows directly from Theorem 3.1. □

Remark 3.1. Theorem 3.1 can be easily extended to 1-safe Petri nets where each transition has exactly one input place (for the definition of 1-safe Petri nets see, for example, Jategaonkar and Meyer (1996)). It is enough to introduce for each $\alpha_i/\overline{\alpha_i}$ and $\beta_i/\overline{\beta_i}$, $1 \leqslant i \leqslant n$, a new set of process constants $\{Q_1, \ldots, Q_k\}$ to ensure that in each reachable marking there is at most one token in every place. Related results about 1-safe Petri nets can be found in Jategaonkar and Meyer (1996).

Another problem we will analyse is weak regularity of BPP processes.

Problem:	Weak regularity of (normed) BPP
Instance:	A (normed) BPP process (P, Δ).
Question:	Is there a finite-state process (F, Δ') such that $(P, \Delta) \approx (F, \Delta')$?

Mayr has proved that weak regularity of BPP is Π_2^P-hard (Mayr 2000a), demonstrating a reduction from the weak bisimilarity problem between a pair of special processes with finitely many reachable states. It can be easily seen that his proof also works for a general pair of weakly regular processes and, moreover, it preserves normedness.

Theorem 3.2 (Mayr 2000a). Let (P_1, Δ) and (P_2, Δ) be weakly regular BPP processes. We can construct in polynomial time a BPP process (P, Δ') such that

$$(P_1, \Delta) \approx (P_2, \Delta) \iff (P, \Delta') \text{ is weakly regular.}$$

Moreover, if (P_1, Δ) and (P_2, Δ) are normed, so is (P, Δ').

Observe that the processes P_1 and P_2 from the proof of *PSPACE*-hardness of weak bisimilarity (Theorem 3.1) are regular and, moreover, they are normed. This gives the following theorem with an immediate corollary.

Theorem 3.3. Weak regularity of normed BPP is *PSPACE*-hard.

Proof. Because of Theorem 3.2, there is a reduction from a *PSPACE*-hard problem of weak bisimilarity for normed BPP to weak regularity of normed BPP. □

Corollary 3.2. Weak regularity of BPP is *PSPACE*-hard.

4. Hardness of weak bisimilarity and regularity for BPA

In this section we consider the same problems for BPA as we did for BPP.

Problem:	Weak bisimilarity of (normed) BPA
Instance:	Two (normed) BPA processes (P_1, Δ) and (P_2, Δ).
Question:	$(P_1, \Delta) \approx (P_2, \Delta)$?

Problem:	Weak regularity of (normed) BPA
Instance:	A (normed) BPA process (P, Δ).
Question:	Is there a finite-state process (F, Δ') such that $(P, \Delta) \approx (F, \Delta')$?

First we show that there is a reduction from weak bisimilarity of regular BPA to weak regularity. The idea of the proof is similar to the case of BPP mentioned above from Mayr (2000a).

Theorem 4.1. Let (P_1, Δ) and (P_2, Δ) be weakly regular BPA processes. We can construct in polynomial time a BPA process (P, Δ') such that

$$(P_1, \Delta) \approx (P_2, \Delta) \iff (P, \Delta') \text{ is weakly regular.}$$

Moreover, if (P_1, Δ) and (P_2, Δ) are normed, so is (P, Δ').

Proof. Assume that (P_1, Δ) and (P_2, Δ) are weakly regular BPA processes. We construct a BPA process (P, Δ') with

$$\mathscr{C}onst(\Delta') \stackrel{\text{def}}{=} \mathscr{C}onst(\Delta) \cup \{A, B, C, B_1, B_2\}$$

and

$$\mathscr{A}ct(\Delta') \stackrel{\text{def}}{=} \mathscr{A}ct(\Delta) \cup \{a\}$$

where A, B, C, B_1, B_2 are new process constants and a is a new action. Then $\Delta' \stackrel{\text{def}}{=} \Delta \cup \Delta^1 \cup \Delta^2$, where Δ^1 and Δ^2 are defined as follows. The set of transition rules Δ^1 is given by

$$\begin{array}{ll} A \xrightarrow{a} A.B & A \xrightarrow{\tau} \epsilon \\ B \xrightarrow{a} \epsilon & B \xrightarrow{\tau} \epsilon \\ & \\ C \xrightarrow{a} B_1 & C \xrightarrow{a} P_1 \\ B_1 \xrightarrow{a} B_1 & B_1 \xrightarrow{a} P_1, \end{array}$$

and Δ^2 is given by

$$\begin{array}{ll} C \xrightarrow{a} B_2 & C \xrightarrow{a} P_2 \\ B_2 \xrightarrow{a} B_2 & B_2 \xrightarrow{a} P_2. \end{array}$$

Let $P \stackrel{\text{def}}{=} A.C$. Observe that if (P_1, Δ) and (P_2, Δ) are normed, so is (P, Δ'). We show now that our reduction is correct.

Lemma 4.1. If $(P_1, \Delta) \not\approx (P_2, \Delta)$, then (P, Δ') is not weakly regular.

Proof. Suppose that $(P_1, \Delta) \not\approx (P_2, \Delta)$. Then we demonstrate that there are infinitely many weakly non-bisimilar states reachable from P. Let us consider $B^i.C$ for any natural number i. Of course, $P \longrightarrow^* B^i.C$, and we claim that $(B^i.C, \Delta') \not\approx (B^j.C, \Delta')$ for any $i \neq j$. Without loss of generality, assume that $i < j$. The attacker has the following winning strategy (playing only in the second process – see Figure 3). He performs a sequence of j actions a in $B^j.C$, thus reaching C. Since B^i cannot do this sequence, the defender has

$$\begin{tikzcd}
B^i.C \arrow[d, "a^{i'}"] \arrow[r, phantom, "\approx^?"] & B^j.C \arrow[d, "a^{i'}"] \\
C \arrow[d, "a"] & B^{j-i'}.C \arrow[d, "a"] \\
B_1 \arrow[d, "a^{j-i'-1}"] & B^{j-i'-1}.C \arrow[d, "a^{j-i'-1}"] \\
B_1 \arrow[d, "a"] & C \arrow[d, "a"] \\
P_1 \arrow[r, phantom, "\not\approx"] & P_2
\end{tikzcd}$$

Fig. 3. The winning strategy for the attacker ($i < j$).

to reach C eventually (let us say after i' steps, where $i' \leqslant i$). As neither P_1 nor P_2 can perform a, he has only two choices when responding to the action a – either $C \xrightarrow{a} B_1$ or $C \xrightarrow{a} B_2$. Assume that he chooses B_1 (the other case is symmetric). Now the defender's only possibility is to stay in B_1 for another $a^{j-i'-1}$ moves of the attacker. After the attacker has reached C (in the second process), he chooses to go to P_2 in the next round. If the defender stays in B_1, he loses immediately, and if he moves to P_1, he loses as well, since $(P_1, \Delta') \not\approx (P_2, \Delta')$. $\square$

Lemma 4.2. If $(P_1, \Delta) \approx (P_2, \Delta)$, then (P, Δ') is weakly regular.

Proof. Assume that $(P_1, \Delta) \approx (P_2, \Delta)$, which implies that $(P, \Delta') \approx (P, \Delta'')$, where $\Delta'' = \Delta' \smallsetminus \Delta^2$ (weak bisimilarity is a congruence on BPA). Notice that (B_1, Δ'') is weakly regular, so it is enough to show that $(A.C, \Delta'') \approx (B_1, \Delta'')$. Obviously, $(C, \Delta'') \approx (B_1, \Delta'')$, which implies that for any $n \geqslant 0$, $(B^n.C, \Delta'') \approx (B_1, \Delta'')$ since $B_1 \xrightarrow{a} B_1$ and $B^n \xrightarrow{\tau^*} \epsilon$. This gives $(A.B^n.C, \Delta'') \approx (B_1, \Delta'')$ for any $n \geqslant 0$, which, in particular, means that $(A.C, \Delta'') \approx (B_1, \Delta'')$. $\square$

Theorem 4.1 is now an immediate consequence of Lemmas 4.1 and 4.2. $\square$

Stribrna showed in Stribrna (1998b, Theorem 2.5) that weak bisimilarity for totally normed BPA is *NP*-hard. The proof is by reduction from a variant of the bin-packing (knapsack) problem, and the processes in this proof have finitely many reachable states (and are thus weakly regular). Thus we can use Theorem 4.1 to obtain the following result with an obvious corollary.

Theorem 4.2. Weak regularity of normed BPA is *NP*-hard.

Corollary 4.1. Weak regularity of BPA is *NP*-hard.

We remind the reader of the fact that *PSPACE*-hardness of weak bisimilarity for BPA achieved by Stribrna (Stribrna 1998b) does not imply *PSPACE*-hardness of weak regularity for BPA, since the described processes are not regular. In the next theorem, however, we prove that weak regularity for BPA is not only *NP*-hard but also *co-NP*-hard. This we demonstrate by showing that weak bisimilarity for BPA is *co-NP*-hard, where the processes involved are finite-state (nevertheless, they are unnormed in this case).

Theorem 4.3. Weak regularity of BPA is *co-NP*-hard.

Proof. We reduce the complement of 3-SAT (Papadimitriou 1994) to weak bisimilarity of weakly regular BPA, and then we use Theorem 4.1.

Problem:	3-SAT COMPLEMENT
Instance:	A natural number n and a Boolean formula ϕ in disjunctive normal form with implicants of length 3 and with Boolean variables $x_1, \ldots, x_n$.
Question:	Is $\forall x_1 \forall x_2 \ldots \forall x_n . \phi$ true?

Let

$$D \equiv \forall x_1 \forall x_2 \ldots \forall x_n . D_1 \vee D_2 \vee \ldots \vee D_k$$

be an instance of 3-SAT COMPLEMENT, where each implicant D_j, $1 \leqslant j \leqslant k$, is a conjunction of three literals. Let us define the following BPA processes (X_1, Δ) and (X_1', Δ), where

$$\begin{aligned} \mathscr{C}onst(\Delta) \stackrel{\text{def}}{=} \big\{ & D_1^1, \ldots, D_k^1, D_1^2, \ldots, D_k^2, D_1^3, \ldots, D_k^3, \\ & X_1, \ldots, X_n, X_{n+1}, X_1', \ldots, X_n', X_{n+1}', Y_1, \ldots, Y_k, A, S \big\} \end{aligned}$$

and

$$\mathscr{A}ct(\Delta) \stackrel{\text{def}}{=} \big\{ d_1^1, \ldots, d_k^1, d_1^2, \ldots, d_k^2, d_1^3, \ldots, d_k^3, x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n, a, s \big\}.$$

For each i, $1 \leqslant i \leqslant n$, let:

α_i be a sequential composition (in some fixed ordering) of process constants D_j^r ($1 \leqslant r \leqslant 3$ and $1 \leqslant j \leqslant k$) such that:

- D_j^1 appears in α_i iff the literal x_i occurs in D_j in the first position
- D_j^2 appears in α_i iff the literal x_i occurs in D_j in the second position
- D_j^3 appears in α_i iff the literal x_i occurs in D_j in the third position;

$\overline{\alpha_i}$ be a sequential composition (in some fixed ordering) of process constants D_j^r ($1 \leqslant r \leqslant 3$ and $1 \leqslant j \leqslant k$) such that:

- D_j^1 appears in $\overline{\alpha_i}$ iff the literal $\neg x_i$ occurs in D_j in the first position
- D_j^2 appears in $\overline{\alpha_i}$ iff the literal $\neg x_i$ occurs in D_j in the second position
- D_j^3 appears in $\overline{\alpha_i}$ iff the literal $\neg x_i$ occurs in D_j in the third position.

The set of transition rules Δ is given by:

$$
\begin{array}{lll}
X_i \xrightarrow{x_i} X_{i+1}.\alpha_i & X'_i \xrightarrow{x_i} X'_{i+1}.\alpha_i & \text{for } 1 \leqslant i \leqslant n \\
X_i \xrightarrow{\overline{x}_i} X_{i+1}.\overline{\alpha_i} & X'_i \xrightarrow{\overline{x}_i} X'_{i+1}.\overline{\alpha_i} & \text{for } 1 \leqslant i \leqslant n \\
X_{n+1} \xrightarrow{a} Y_j & X'_{n+1} \xrightarrow{a} Y_j & \text{for } 1 \leqslant j \leqslant k \\
 & X'_{n+1} \xrightarrow{a} A &
\end{array}
$$

$$
\begin{array}{l}
A \xrightarrow{a} A \\
A \xrightarrow{\tau} \epsilon
\end{array}
$$

$$
S \xrightarrow{s} S
$$

$$
\begin{array}{llll}
Y_j \xrightarrow{d_j^1} S & Y_j \xrightarrow{d_j^2} S & Y_j \xrightarrow{d_j^3} S & \text{for } 1 \leqslant j \leqslant k \\
 & Y_j \xrightarrow{a} Y_j & & \text{for } 1 \leqslant j \leqslant k \\
 & Y_j \xrightarrow{\tau} \epsilon & & \text{for } 1 \leqslant j \leqslant k
\end{array}
$$

$$
\begin{array}{llll}
D_j^1 \xrightarrow{d_j^1} S & D_j^2 \xrightarrow{d_j^2} S & D_j^3 \xrightarrow{d_j^3} S & \text{for } 1 \leqslant j \leqslant k \\
D_j^1 \xrightarrow{\tau} \epsilon & D_j^2 \xrightarrow{\tau} \epsilon & D_j^3 \xrightarrow{\tau} \epsilon & \text{for } 1 \leqslant j \leqslant k.
\end{array}
$$

The intuition is that the attacker plays in X'_1 and generates some truth assignment. When he reaches the process constant A, the defender chooses an implicant that is satisfied by the truth assignment by performing a transition $X_{n+1} \xrightarrow{a} Y_j$. The attacker can now test whether this implicant is indeed satisfied.

Lemma 4.3. If $(X_1, \Delta) \approx (X'_1, \Delta)$, then D is true.

Proof. In order to show a contradiction, suppose that D is false, that is, there is some assignment of truth values for $x_1, \ldots, x_n$ such that $D_1 \vee D_2 \vee \ldots \vee D_k$ is false, which means that for each j, $1 \leqslant j \leqslant k$, there is at least one false literal in D_j. We show that the attacker has a winning strategy in the bisimulation game. First, the attacker plays in X'_1 generating this false assignment, and, finally, he uses the transition $X'_{n+1} \xrightarrow{a} A$. The defender can only respond by performing the same actions $x_i/\overline{x}_i$ with the final transition $X_{n+1} \xrightarrow{a} Y_j$ for some j (observe that the defender cannot use the transition $Y_j \xrightarrow{\tau} \epsilon$, otherwise the attacker wins immediately). Now the attacker changes the processes and plays $Y_j \xrightarrow{d_j^r} S$, where r is a position of a false literal in D_j. This means that the defender loses, since he has no response to this move. □

Note that the winning strategy for the attacker in the proof above requires only one switching of sides (from the side of X'_1 to the side of X_1).

Lemma 4.4. If D is true, then $(X_1, \Delta) \approx (X'_1, \Delta)$.

Proof. We show that the defender has a winning strategy. Whatever the attacker performs during the first n moves, the defender imitates in the other process. Finally, we get a pair of processes $X_{n+1}.\alpha$ and $X'_{n+1}.\alpha$. If the attacker chooses the rule $X_{n+1} \xrightarrow{a} Y_j$ for some j, then he loses, since the defender can do the same move in $X'_{n+1}.\alpha$ and make the resulting processes equal. The same happens if the attacker chooses the rule $X'_{n+1} \xrightarrow{a} Y_j$ for some j in the second process. So the only possibility for the attacker to win is to move under a to $A.\alpha$ in the second process. The defender answers by performing $X_{n+1} \xrightarrow{a} Y_j$, where D_j is an implicant that makes the formula $D_1 \vee D_2 \vee \ldots \vee D_k$ true. Now the attacker has to switch processes, since if he continues in $A.\alpha$ doing the τ action, he loses again (the defender can make the two processes equal). In the process $Y_j.\alpha$ the attacker has essentially two possibilities. He can perform $Y_j \xrightarrow{d_j^r} S$ for some r, $1 \leqslant r \leqslant 3$. However, the defender can perform some sequence of τ actions to enable d_j^r in the second process and then he performs the transition $D_j^r \xrightarrow{d_j^r} S$. As S is unnormed, the resulting processes are bisimilar (since $(S.\beta, \Delta) \approx (S.\beta', \Delta)$ for any β and β'). The other possibility for the attacker is to perform $Y_j \xrightarrow{\tau} \epsilon$ first, but then he again loses (the resulting processes can be made equal). Thus the defender has a winning strategy, which means that $(X_1, \Delta) \approx (X'_1, \Delta)$. □

The proof of Theorem 4.3 is then a consequence of Lemmas 4.3 and 4.4, Theorem 4.1, and the fact that both (X_1, Δ) and (X'_1, Δ) are finite-state processes. □

Corollary 4.1 and Theorem 4.3 show that weak regularity for BPA is both *NP* and *co-NP*-hard. We use these results to obtain *DP*-hardness. The class *DP* is defined as follows (Papadimitriou 1994). A language L is in *DP* iff there are two languages $L_1 \in NP$ and $L_2 \in co\text{-}NP$ such that $L = L_1 \cap L_2$. Obviously, $NP \cup co\text{-}NP$ is contained in *DP* and, moreover, the other inclusion is unlikely. We show that weak regularity is *DP*-hard by demonstrating a reduction from the SAT-UNSAT problem (Papadimitriou 1994).

Problem:	SAT-UNSAT
Instance:	Two Boolean formulas ϕ_1 and ϕ_2.
Question:	Is ϕ_1 satisfiable and ϕ_2 is not?

Theorem 4.4. Weak regularity of BPA is *DP*-hard.

Proof. As we know that weak regularity is both *NP* and *co-NP*-hard, we can in polynomial time construct processes (P_1, Δ) and (P_2, Δ) such that (P_1, Δ) is weakly regular iff ϕ_1 is satisfiable, and (P_2, Δ) is weakly regular iff ϕ_2 is not satisfiable. Let us now construct a process (P, Δ') such that (P, Δ') is weakly regular iff ϕ_1 is satisfiable and ϕ_2 is not. We define $\mathscr{C}onst(\Delta') \stackrel{\text{def}}{=} \mathscr{C}onst(\Delta) \cup \{P\}$ and $\mathscr{A}ct(\Delta') \stackrel{\text{def}}{=} \mathscr{A}ct(\Delta) \cup \{a_1, a_2\}$ where P is a new process constant and a_1, a_2 are new actions. The set Δ' contains all the rules from Δ together with

$$P \xrightarrow{a_1} P_1 \qquad P \xrightarrow{a_2} P_2.$$

Obviously, (P, Δ') is regular iff both (P_1, Δ) and (P_2, Δ) are regular. This proves that (P, Δ') is weakly regular iff ϕ_1 is satisfiable and ϕ_2 is not. □

5. Hardness of strong bisimilarity for BPP

In this section we give a simple proof of Theorem 4 from Mayr (2000a).

Problem:	Strong bisimilarity of BPP
Instance:	Two BPP processes (P_1, Δ) and (P_2, Δ).
Question:	$(P_1, \Delta) \sim (P_2, \Delta)$?

Theorem 5.1. Strong bisimilarity of BPP is *co-NP*-hard.

Proof. We establish a reduction from 3-SAT COMPLEMENT in a way similar to that in the proof of Theorem 4.3. Let

$$D \equiv \forall x_1 \forall x_2 \ldots \forall x_n . D_1 \vee D_2 \vee \ldots \vee D_k$$

be an instance of 3-SAT COMPLEMENT, where each implicant D_j, $1 \leqslant j \leqslant k$, is a conjunction of three literals. We define BPP processes (X_1, Δ) and (X_1', Δ), where

$$\begin{aligned}\mathscr{C}onst(\Delta) \stackrel{\text{def}}{=} \{ & D_1^1, \ldots, D_k^1, D_1^2, \ldots, D_k^2, D_1^3, \ldots, D_k^3, \\ & X_1, \ldots, X_n, X_{n+1}, X_1', \ldots, X_n', X_{n+1}', Y_1, \ldots, Y_k \}\end{aligned}$$

and

$$\mathscr{A}ct(\Delta) \stackrel{\text{def}}{=} \{ d_1^1, \ldots, d_k^1, d_1^2, \ldots, d_k^2, d_1^3, \ldots, d_k^3, x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n, a \}.$$

For each i, $1 \leqslant i \leqslant n$, let α_i and $\overline{\alpha_i}$ have the same definition as in the proof of Theorem 4.3. The only difference is that we replace the sequential composition with the parallel one.

The set of transition rules Δ is given by:

$$\begin{array}{lll} X_i \xrightarrow{x_i} X_{i+1} \| \alpha_i & X_i' \xrightarrow{x_i} X_{i+1}' \| \alpha_i & \text{for } 1 \leqslant i \leqslant n \\ X_i \xrightarrow{\overline{x}_i} X_{i+1} \| \overline{\alpha_i} & X_i' \xrightarrow{\overline{x}_i} X_{i+1}' \| \overline{\alpha_i} & \text{for } 1 \leqslant i \leqslant n \\ X_{n+1} \xrightarrow{a} Y_j & X_{n+1}' \xrightarrow{a} Y_j & \text{for } 1 \leqslant j \leqslant k \\ & X_{n+1}' \xrightarrow{a} \epsilon & \end{array}$$

$$\begin{array}{llll} Y_j \xrightarrow{d_j^1} Y_j & Y_j \xrightarrow{d_j^2} Y_j & Y_j \xrightarrow{d_j^3} Y_j & \text{for } 1 \leqslant j \leqslant k \\ D_j^1 \xrightarrow{d_j^1} D_j^1 & D_j^2 \xrightarrow{d_j^2} D_j^2 & D_j^3 \xrightarrow{d_j^3} D_j^3 & \text{for } 1 \leqslant j \leqslant k. \end{array}$$

Lemma 5.1. If $(X_1, \Delta) \sim (X_1', \Delta)$, then D is true.

Proof. In order to show a contradiction, suppose that D is false, that is, there is some assignment of truth values for $x_1, \ldots, x_n$ such that $D_1 \vee D_2 \vee \ldots \vee D_k$ is false, which means that for each j, $1 \leqslant j \leqslant k$, there is at least one false literal in D_j. We show that the attacker has a winning strategy in the bisimulation game. First, the attacker plays in X'_1 generating this false assignment, and, finally, he uses the transition $X'_{n+1} \xrightarrow{a} \epsilon$. The defender can only respond by performing the same actions $x_i/\overline{x}_i$ with the final transition $X_{n+1} \xrightarrow{a} Y_j$ for some j. Now the attacker changes the processes and plays $Y_j \xrightarrow{d_j^r} S$, where r is a position of a false literal in D_j. This means that the defender loses, since he has no response to this move. □

Lemma 5.2. If D is true, then $(X_1, \Delta) \sim (X'_1, \Delta)$.

Proof. We show that the defender has a winning strategy. Whatever the attacker performs during the first n moves the defender imitates in the other process (note that the attacker cannot win by performing d_j^m for some $1 \leqslant m \leqslant 3$ and $1 \leqslant j \leqslant k$ until X_{n+1} and X'_{n+1} are reached). Finally, we get a pair of processes $X_{n+1}\|\alpha$ and $X'_{n+1}\|\alpha$. If the attacker chooses the rule $X_{n+1} \xrightarrow{a} Y_j$ for some j, he loses, since the defender can do the same move in $X'_{n+1}\|\alpha$ and make the resulting processes equal. The same happens if the attacker chooses the rule $X'_{n+1} \xrightarrow{a} Y_j$ for some j in the second process. So, the only possibility for the attacker to win is to move under a to α in the second process. The defender answers by performing $X_{n+1} \xrightarrow{a} Y_j$, where D_j is the implicant that makes the formula $D_1 \vee D_2 \vee \ldots \vee D_k$ true. However, since D_j is true, $Y_j\|\alpha \sim \alpha$. So the defender has a winning strategy, which means that $(X_1, \Delta) \sim (X'_1, \Delta)$. □

Thus the problem of strong bisimilarity for BPP is *co-NP*-hard because of Lemmas 5.1 and 5.2. □

6. Conclusion

In the following tables we summarise the known results for weak bisimilarity and regularity problems for BPA, BPP and PA. The results obtained in this paper are in boldface. A question mark means that no lower bound is known yet.

	Weak bisimilarity	Weak bisimilarity of normed processes
BPA	*PSPACE*-hard (Stribrna 1998b)	*NP*-hard (Stribrna 1998b)
BPP	*NP*-hard (Stribrna 1998b) Π_2^P-hard (Mayr 2000a) **PSPACE-hard**	*NP*-hard (Stribrna 1998b) **PSPACE-hard**
PA	*PSPACE*-hard (Stribrna 1998b) **PSPACE-hard**	*NP*-hard (Stribrna 1998b) **PSPACE-hard**

For the case of weak bisimilarity in the class of PA, the result in this paper is more general, since our processes are weakly regular, which is not the case for the result by Stribrna.

	Weak regularity	Weak regularity of normed processes
BPA	? **DP-hard**	? **NP-hard**
BPP	Π_2^p-hard (Mayr 2000a) **PSPACE-hard**	? **PSPACE-hard**
PA	Π_2^p-hard (Mayr 2000a) **PSPACE-hard**	? **PSPACE-hard**

Remember that *DP*-hardness means, in particular, both *NP* and *co-NP*-hardness. Our results could indicate that (unlike the case of strong bisimilarity) there is not much difference between the complexity of weak bisimilarity/regularity for normed and unnormed processes – for example, weak bisimilarity and regularity is *PSPACE*-hard for both normed and unnormed BPP. However, we still use only a little of the power of bisimilarity since in our proofs the attacker switches the sides of processes at most once. It is possible that cleverer reductions may improve the lower bounds by exploiting the possibility of switching sides in the bisimulation game an arbitrary number of times. This could also show a substantial difference between the normed and unnormed case, which is known to be delicate when considering strong bisimilarity.

Final remark

After the acceptance of this paper, the author further developed some of the techniques presented here and achieved a *PSPACE* lower bound for strong bisimilarity and strong regularity of BPP (Srba 2002), thus improving upon Theorem 5.1. He also claims that more involved techniques can be used to show *PSPACE*-hardness of strong bisimilarity and strong regularity of BPA, which pushes the *DP* lower bound of weak regularity for BPA to *PSPACE*.

Acknowledgement

I would like to thank my advisor Mogens Nielsen for his kind supervision and encouragement, and the journal referees for useful comments.

References

Baeten, J. C. M., Bergstra, J. A. and Klop, J. W. (1987) On the consistency of Koomen's fair abstraction rule. *Theoretical Computer Science* **51** (1–2) 129–176.

Baeten, J. C. M. and Weijland, W. P. (1990) *Process Algebra*. Cambridge Tracts in Theoretical Computer Science **18**, Cambridge University Press.

Bouajjani, A., Esparza, J. and Maler, O. (1997) Reachability analysis of pushdown automata: Application to model-checking. In: CONCUR '97: Concurrency Theory, 8th International Conference. *Springer-Verlag Lecture Notes in Computer Science* **1243** 135–150.

Burkart, O., Caucal, D. and Steffen, B. (1995) An elementary decision procedure for arbitrary context-free processes. In: Proceedings of MFCS'95. *Springer-Verlag Lecture Notes in Computer Science* **969** 423–433.

Burkart, O., Caucal, D. and Steffen, B. (1996) Bisimulation collapse and the process taxonomy. In: Proceedings of CONCUR'96. *Springer-Verlag Lecture Notes in Computer Science* **1119** 247–262.

Burkart, O. and Esparza, J. (1997) More infinite results. *Bulletin of the European Association for Theoretical Computer Science* **62** 138–159.

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

Christensen, S., Hüttel, H. and Stirling, C. (1995) Bisimulation equivalence is decidable for all context-free processes. *Information and Computation* **121** 143–148.

Esparza, J. (1995) Petri nets, commutative context-free grammars, and basic parallel processes. In: Proceedings of FCT'95. *Springer-Verlag Lecture Notes in Computer Science* **965** 221–232.

Hirshfeld, Y. (1996) Bisimulation trees and the decidability of weak bisimulations. In: Proceedings of the the First International Workshop on Verification of Infinite State Systems (Infinity'96). *ENTCS* **5**, Springer-Verlag.

Hirshfeld, Y., Jerrum, M. and Moller, F. (1996a) A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theoretical Computer Science* **158** (1–2) 143–159.

Hirshfeld, Y., Jerrum, M. and Moller, F. (1996b) A polynomial-time algorithm for deciding bisimulation equivalence of normed basic parallel processes. *Mathematical Structures in Computer Science* **6** (3) 251–259.

Jancar, P. and Esparza, J. (1996) Deciding finiteness of Petri nets up to bisimilarity. In: Proceedings of ICALP'96. *Springer-Verlag Lecture Notes in Computer Science* **1099** 478–489.

Jancar, P., Kucera, A. and Mayr, R. (1998) Deciding bisimulation-like equivalences with finite-state processes. In: Proceedings of the Annual International Colloquium on Automata, Languages and Programming (ICALP'98). *Springer-Verlag Lecture Notes in Computer Science* **1443**.

Jategaonkar, L. and Meyer, A. R. (1996) Deciding true concurrency equivalences on safe, finite nets. *Theoretical Computer Science* **154** (1) 107–143.

Kucera, A. (1996) Regularity is decidable for normed PA processes in polynomial time. In: Proceedings of FST&TCS'96. *Springer-Verlag Lecture Notes in Computer Science* **1180** 111–122.

Kucera, A. and Mayr, R. (1999) Weak bisimilarity with infinite-state systems can be decided in polynomial time. In: Proceedings of the 10th International Conference on Concurrency Theory (CONCUR'99). *Springer-Verlag Lecture Notes in Computer Science* **1664**.

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

Mayr, R. (2000a) On the complexity of bisimulation problems for basic parallel processes. In: Proceedings of 27st International Colloquium on Automata, Languages and Programming (ICALP'00). *Springer-Verlag Lecture Notes in Computer Science* **1853** 329–341.

Mayr, R. (2000b) On the complexity of bisimulation problems for pushdown automata. In: IFIP International Conference on Theoretical Computer Science (IFIP TCS'2000). *Springer-Verlag Lecture Notes in Computer Science* **1872**.

Mayr, R. (2000c) Process rewrite systems. *Information and Computation* **156** (1) 264–286.

Milner, R. (1989) *Communication and Concurrency*, Prentice-Hall.

Moller, F. (1996) Infinite results. In: Proceedings of CONCUR'96. *Springer-Verlag Lecture Notes in Computer Science* **1119** 195–216.

Papadimitriou, C. H. (1994) *Computational Complexity*, Addison-Wesley.

Park, D. M. R. (1981) Concurrency and automata on infinite sequences. In: Proceedings 5th GI Conference. *Springer-Verlag Lecture Notes in Computer Science* **104** 167–183.

Plotkin, G. (1981) A structural approach to operational semantics. Technical Report Daimi FN-19, Department of Computer Science, University of Aarhus.

Srba, J. (2002) Strong bisimilarity and regularity of basic parallel processes is PSPACE-hard. In: Proceedings of the 19th International Symposium on Theoretical Aspects of Computer Science (STACS'02). *Springer-Verlag Lecture Notes in Computer Science* **2285** 535–546.

Stirling, C. (1995) Local model checking games. In: Proceedings of the 6th International Conference on Concurrency Theory (CONCUR'95). *Springer-Verlag Lecture Notes in Computer Science* **962** 1–11.

Stirling, C. (2001) Decidability of weak bisimilarity for a subset of basic parallel processes. In: Proceedings of the 4th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'01). *Springer-Verlag Lecture Notes in Computer Science* **2030** 379–393.

Stribrna, J. (1998a) *Decidability and complexity of equivalences on simple process algebras*, Ph.D. thesis, University of Edinburgh.

Stribrna, J. (1998b) Hardness results for weak bisimilarity of simple process algebras. In: Proceedings of the MFCS'98 Workshop on Concurrency. *ENTCS* **18**, Springer-Verlag.

Stribrna, J. (1999) Approximating weak bisimulation on basic process algebras. In: Proceedings of 24rd International Symposium on Mathematical Foundations of Computer Science (MFCS'99). *Springer-Verlag Lecture Notes in Computer Science* **1672** 336–375.

Thomas, W. (1993) 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). *Springer-Verlag Lecture Notes in Computer Science* **668** 559–568.

Walukiewicz, I. (1996) Pushdown processes: Games and model checking. In: International Conference on Computer-Aided Verification (CAV'96). *Springer-Verlag Lecture Notes in Computer Science* **1102** 62–74. (Also to appear in *Information and Computation*.)