On the Complexity of Bisimulation Problems for Basic Parallel Processes

Richard Mayr

LIAFA - Université Denis Diderot - Case 7014 - 2, place Jussieu,
F-75251 Paris Cedex 05. France. E-mail: mayr@liafa.jussieu.fr

Abstract. Strong bisimilarity of Basic Parallel Processes (BPP) is decidable, but the best known algorithm has non-elementary complexity [7]. On the other hand, no lower bound for the problem was known. We show that strong bisimilarity of BPP is co-$\mathcal{NP}$-hard.
Weak bisimilarity of BPP is not known to be decidable, but an $\mathcal{NP}$ lower bound has been shown in [31]. We improve this result by showing that weak bisimilarity of BPP is Π_2^p-hard.
Finally, we show that the problems if a BPP is regular (i.e., finite) w.r.t. strong and weak bisimilarity are co-$\mathcal{NP}$-hard and Π_2^p-hard, respectively.

1 Introduction

Bisimulation equivalence plays a central role in the theory of process algebras [25]. The decidability and complexity of bisimulation problems for infinite-state systems has been studied intensively (see [26] for a survey). While many algorithms for bisimulation problems have a very high complexity, only few lower bounds are known.

The state of the art. Strong bisimilarity of two Petri nets and weak bisimilarity of a Petri net and a finite automaton is undecidable [13, 14]. Weak bisimilarity for Basic Parallel Processes (BPP) is $\mathcal{NP}$-hard and weak bisimilarity for context-free processes (BPA) is $PSPACE$-hard [31]. However, it is still an open question whether these problems are decidable.

Some lower bounds for decidable bisimulation problems have been shown in [23]. Strong (and weak) bisimilarity between pushdown automata (PDA) and finite automata is $PSPACE$-hard, finiteness of PDA w.r.t. weak and strong bisimilarity also $PSPACE$-hard. Finally, both strong bisimilarity of Petri nets and finite automata and finiteness of Petri nets w.r.t. strong bisimilarity are $EXPSPACE$-hard. (See the table in Section 5 for a summary of all results on the complexity of bisimulation problems.)

Basic Parallel Processes (BPP) were introduced by Christensen [6] as the fragment of CCS [25] without communication, restriction and relabeling. They are equivalent to communication-free nets [8], the subclass of Petri nets [28] where every transition has exactly one input-place with arc-weight one. While strong (and weak) bisimilarity are undecidable for Petri nets [13], strong bisimilarity is decidable for BPP (i.e., communication-free nets) [7]. However, the algorithm in [7] has non-elementary complexity and, to the best of our knowledge, no better algorithm has been found since then. In spite of this, no lower bound for the problem has been found either.

However, there is a polynomial algorithm for bisimilarity on the restricted subclass of *normed* BPP [12]. (A process is *normed* iff from every reachable state there is a terminating computation.) Thus, it was conjectured that a polynomial algorithm should also exist for general (unnormed) BPP. This belief was reinforced by the fact that many other problems for BPP are polynomial: boundedness [17], termination, liveness, (partial) deadlock reachability and (partial) livelock reachability [21, 22]. (On the other hand there are also hard problems for BPP: reachability is $\mathcal{NP}$-complete [8], some model checking problems are $PSPACE$-complete [20, 22] or even undecidable [9].)

Our contribution. We show that strong bisimilarity for BPP is co-$\mathcal{NP}$-hard (thus proving the above mentioned conjecture wrong). We also show that weak bisimilarity for BPP is Π_2^p-hard, thus improving a previously established $\mathcal{NP}$ lower bound [31]. Finally, we show that the problem if a BPP is regular (i.e., finite) w.r.t. strong and weak bisimilarity is co-$\mathcal{NP}$-hard and Π_2^p-hard, respectively.

2 Definitions

Let $Act = \{a, b, c, \ldots\}$ and $Const = \{\epsilon, X, Y, Z, \ldots\}$ be disjoint countably infinite sets of *actions* and *process constants*, respectively. The class of *general process expressions* G is defined by $E ::= \epsilon \mid X \mid E\|E \mid E.E$, where $X \in Const$ and ϵ is a special constant that denotes the empty expression. Intuitively, '.' is a sequential composition and '$\|$' is a parallel composition. We do not distinguish between expressions related by *structural congruence* which is given by the following laws: '.' and '$\|$' are associative, '$\|$' is commutative, and 'ϵ' is a unit for '.' and '$\|$'.

A *process rewrite system* (PRS) [24] is specified by a finite set Δ of *rules* which have the form $E \stackrel{a}{\rightarrow} F$, where $E, F \in G$, $E \neq \epsilon$ and $a \in Act$. $Const(\Delta)$ and $Act(\Delta)$ denote the sets of process constants and actions which are used in the rules of Δ, respectively (note that these sets are finite). Each process rewrite system Δ defines a unique transition system where states are process expressions over $Const(\Delta)$. $Act(\Delta)$ is the set of labels. The transitions are determined by Δ and the following inference rules (remember that '$\|$' is commutative):

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

We extend the notation $E \stackrel{a}{\rightarrow} F$ to elements of Act^* in a standard way. Moreover, we say that F is *reachable* from E if $E \stackrel{w}{\rightarrow} F$ for some $w \in Act^*$.

Various subclasses of process rewrite systems can be obtained by imposing certain restrictions on the form of rules. To specify those restrictions, we first define the classes S and P of *sequential* and *parallel* expressions, composed of all process expressions which do not contain the '$\|$' and the '.' operator, respectively. We also use '1' to denote the set of process constants. The hierarchy of process rewrite systems is presented in Fig. 1; the restrictions are specified by a pair (A, B), where A and B are the classes of expressions which can appear on the left-hand and the right-hand side of rules, respectively. This hierarchy contains almost all classes of infinite state systems which have been studied so far; BPA (Basic Process Algebra, also called context-free processes), BPP (Basic Parallel Processes), and PA-processes are well-known [1],

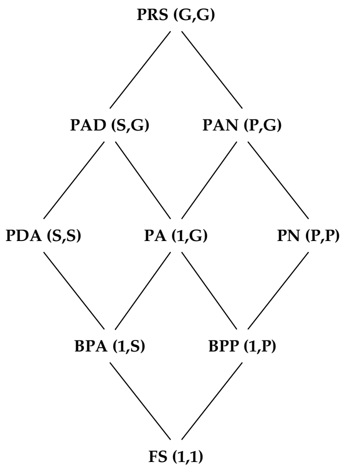

Fig. 1. A hierarchy of PRS

PDA correspond to pushdown automata (as proved by Caucal in [5]), PN correspond to Petri nets, PRS stands for 'Process Rewrite Systems', PAD and PAN are artificial names made by combining existing ones (PAD = PA+PDA, PAN = PA+PN).
Here we study Basic Parallel Processes (BPP) that correspond to process rewrite systems of type $(1, P)$.
We consider the semantical equivalences *weak bisimilarity* and *strong bisimilarity* [25] over transition systems generated by PRS.

Definition 1. *The action τ is a special 'silent' internal action. The* extended transition relation *'$\stackrel{a}{\Rightarrow}$' is defined by $E \stackrel{a}{\Rightarrow} F$ iff either $E = F$ and $a = \tau$, or $E \stackrel{\tau^i}{\to} E' \stackrel{a}{\to} E'' \stackrel{\tau^j}{\to} F$ for some $i, j \in \mathbb{N}_0$, $E', E'' \in G$. A binary relation R over process expressions is a* weak bisimulation *iff whenever $(E, F) \in R$ then for every $a \in Act$: if $E \stackrel{a}{\to} E'$ then there is $F \stackrel{a}{\Rightarrow} F'$ s.t. $(E', F') \in R$ and if $F \stackrel{a}{\to} F'$ then there is $E \stackrel{a}{\Rightarrow} E'$ s.t. $(E', F') \in R$. Processes E, F are* weakly bisimilar*, written $E \approx F$, iff there is a weak bisimulation relating them. Strong bisimulation is defined similarly with $\stackrel{a}{\to}$ instead of $\stackrel{a}{\Rightarrow}$. Processes E, F are* strongly bisimilar*, written $E \sim F$, iff there is a strong bisimulation relating them. The largest (strong or weak) bisimulation is an equivalence relation.*

Bisimulation equivalence can also be described by *bisimulation games* [30, 32] between two players. One player, the 'attacker', tries to prove that two given processes are not bisimilar, while the other player, the 'defender', tries to frustrate this. In every round of the game the attacker chooses one process and performs an action. The defender must imitate this move and perform the same action in the other process (possibly together with several internal τ-actions in the case of weak bisimulation). If one player cannot move then the other player wins. The defender wins every infinite game. Two processes are bisimilar iff the defender has a winning strategy and non-bisimilar iff the attacker has a winning strategy.

3 Hardness of Strong Bisimilarity for BPP

STRONG BISIMILARITY OF BPP
Instance: Two BPP processes P_1 and P_2.
Question: $P_1 \sim P_2$?

This problem has been shown to be decidable in [7]. However, the algorithm relies on Dickson's Lemma for termination and therefore the algorithm is not primitive recursive. A polynomial algorithm for bisimilarity on the restricted subclass of *normed* BPP has been described in [12], which led to the conjecture that the general problem was also polynomial. We prove this conjecture wrong by proving a co-$\mathcal{NP}$ lower bound. Thus no polynomial algorithm for strong bisimilarity of BPP can exist, unless $\mathcal{P} = \mathcal{NP}$.
First we give an intuition why the general (unnormed) problem is so hard, using the terminology of communication-free Petri nets. The problem if a place in a communication-free net is unbounded (i.e., if there are reachable states that put arbitrarily high numbers of tokens on it) is easily decidable in polynomial time [17]. However, it is not so easy to determine if the number of tokens on a place really matters w.r.t. bisimilarity, i.e., if states with different numbers of tokens on this place are really different w.r.t. bisimilarity (i.e., non-bisimilar). First we consider the simple example Δ:

$X \xrightarrow{a} X\|Y$, $X \xrightarrow{a} \epsilon$, $Y \xrightarrow{a} \epsilon$, $Z \xrightarrow{a} Z$. The process (X, Δ) is infinite w.r.t. bisimilarity (since it has infinitely many non-bisimilar reachable states). However, $(X\|Z, \Delta)$ is finite w.r.t. bisimilarity, since $(X\|Z, \Delta) \sim (Z, \Delta)$. We say that in the process $(X\|Z, \Delta)$ the subprocess (Z, Δ) *masks* the infiniteness of (X, Δ). In particular, the subprocess Z has the effect that the number of subprocesses Y doesn't matter for bisimilarity, since $(Y^n\|Z, \Delta) \sim (Y^m\|Z, \Delta)$ for any $n, m \in \mathbb{N}$. Now consider the new system $\Delta' := \Delta \cup \{Z \xrightarrow{a} \epsilon\}$. The process $(X\|Z, \Delta')$ is infinite w.r.t. bisimilarity, because $(X\|Z, \Delta') \xrightarrow{a} (X, \Delta')$. We say that by this transition the subprocess X is *unmasked*. Of course, this is only a very trivial example of masking and unmasking. In general the question if a process can be unmasked (i.e., if a place matters w.r.t. bisimilarity) is $\mathcal{NP}$-hard. Later in this section we use a more complex example of masking and unmasking to prove this.

For the subclass of *normed* BPP, finiteness w.r.t. bisimilarity coincides with boundedness. Thus for *normed* BPP finiteness w.r.t. bisimilarity is decomposable into properties of subprocesses and decidable in polynomial time. In particular, for *normed* BPP, the parallel composition of two infinite processes yields an infinite process. For general BPP it is different. The parallel composition of infinite processes (w.r.t. bisimilarity) can yield a process that is finite w.r.t. bisimilarity. Thus, finiteness (or infiniteness) w.r.t. bisimilarity of a BPP process cannot be decomposed into properties of subprocesses in general. The following example shows this. Let Δ be

$$
\begin{array}{l}
X_i \xrightarrow{a_{i+1}} X_i\|Y_i, \quad Y_i \xrightarrow{a_i} \epsilon \text{ for } 1 \leq i \leq n-1 \\
X_n \xrightarrow{a_1} X_n\|Y_n, \; Y_n \xrightarrow{a_n} \epsilon
\end{array}
$$

Then the process $X_1\|X_2\|\ldots\|X_n$ is finite w.r.t. bisimilarity, but every subprocess (e.g. $X_3\|X_4\|X_7$) is infinite w.r.t. bisimilarity.

Now we are ready to prove the co-$\mathcal{NP}$ lower bound for strong bisimilarity of BPP. We do this by a polynomial reduction of 3-SAT to the negation of the problem. Let $n \in \mathbb{N}$ and let $x_1, \ldots, x_n$ be boolean variables. A literal is either a variable or the negation of a variable. A clause is a disjunction of 3 literals. Let $Q := Q_1 \wedge \ldots \wedge Q_k$ be a boolean formula in 3-CNF over $x_1, \ldots, x_n$ with k clauses. We construct BPPs P_1 and P_2 s.t. Q is satisfiable iff $P_1 \not\sim P_2$. The set of transition rules Δ is defined as follows.

For every $i \in \{1, \ldots, n\}$ we have $X_i \xrightarrow{x_i} X_{i+1}\|\alpha_i$ where α_i is a parallel composition of constants defined as follows: For every $j \in \{1, \ldots, k\}$ let A_j be in α_i iff the first literal of Q_j is $\bar{x}_i$. For every $j \in \{1, \ldots, k\}$ let B_j be in α_i iff the second literal of Q_j is $\bar{x}_i$. For every $j \in \{1, \ldots, k\}$ let C_j be in α_i iff the third literal of Q_j is $\bar{x}_i$. For every $i \in \{1, \ldots, n\}$ we have $X_i \xrightarrow{\bar{x}_i} X_{i+1}\|\beta_i$ where β_i is a parallel composition of constants defined as follows: For every $j \in \{1, \ldots, k\}$ let A_j be in β_i iff the first literal of Q_j is x_i. For every $j \in \{1, \ldots, k\}$ let B_j be in β_i iff the second literal of Q_j is x_i. For every $j \in \{1, \ldots, k\}$ let C_j be in β_i iff the third literal of Q_j is x_i. The intuition is that by action $x_i/\bar{x}_i$ one chooses the value *true*/*false* for the variable x_i. Q is satisfiable iff the assignment of values to the variables can be chosen in such a way that for every $j \in \{1, \ldots, k\}$ at least one of the constants $\{A_j, B_j, C_j\}$ does *not* appear.

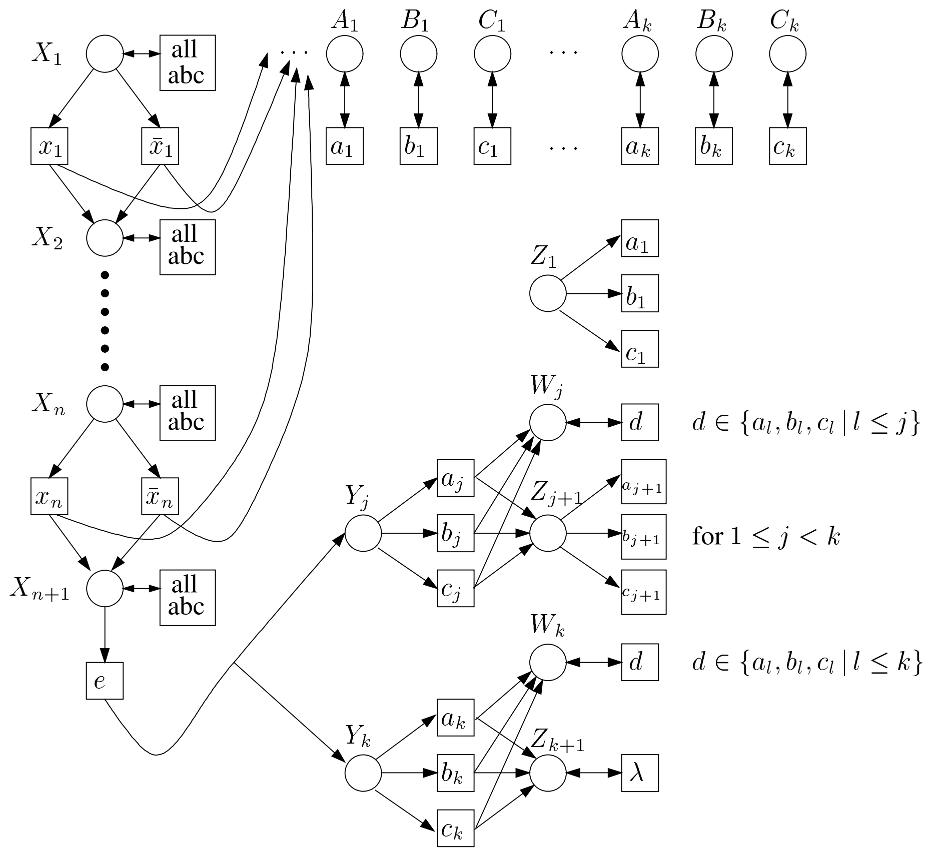

Fig. 2. Δ in Petri net notation. 'All abc' means all actions a_i, b_i, c_i for every $i \in \{1, \ldots, k\}$.

The other transition rules are as follows:

$$
\begin{array}{lll}
A_j & \xrightarrow{a_j} A_j & \text{for } 1 \leq j \leq k \\
B_j & \xrightarrow{b_j} B_j & \text{for } 1 \leq j \leq k \\
C_j & \xrightarrow{c_j} C_j & \text{for } 1 \leq j \leq k \\
X_i & \xrightarrow{d_j} X_i & \text{for } 1 \leq i \leq n+1, 1 \leq j \leq k, d_j \in \{a_j, b_j, c_j\} \\
X_{n+1} & \xrightarrow{e} Y_1 \| \ldots \| Y_k & \\
Y_j & \xrightarrow{d_j} Z_{j+1} \| W_j & \text{for } 1 \leq j \leq k, d_j \in \{a_j, b_j, c_j\} \\
W_j & \xrightarrow{d_l} W_j & \text{for } 1 \leq j \leq k, 1 \leq l \leq j, d_l \in \{a_l, b_l, c_l\} \\
Z_j & \xrightarrow{d_j} \epsilon & \text{for } 1 \leq j \leq k, d_j \in \{a_j, b_j, c_j\} \\
Z_{k+1} & \xrightarrow{\lambda} Z_{k+1} &
\end{array}
$$

Figure 2 gives a rough description of Δ in Petri net notation. Let $P_1 := (X_1 \| Z_1, \Delta)$ and $P_2 := (X_1, \Delta)$.

Lemma 2. *If Q is satisfiable then $P_1 \not\sim P_2$.*

Proof. We show that the attacker has the following winning strategy. Since Q is satisfiable, there exists an assignment of variables that makes Q true. The attacker can choose this assignment by performing the corresponding actions x_i or $\bar{x}_i$ for $1 \le i \le n$ in either P_1 or P_2. Then the attacker does the action e. The defender can only respond by doing exactly the same. This yields the new states P_1' and P_2' with $P_1' = P_2' \| Z_1$. For every $j \in \{1, \dots, k\}$ there is at least one constant $D_j \in \{A_j, B_j, C_j\}$ that does not appear in P_1' or P_2'. Let d_j be the action corresponding to D_j, e.g. if $D_1 = B_1$ then $d_1 = b_1$.

The attacker performs the action d_1 by the rule $Z_1 \stackrel{d_1}{\to} \epsilon$ in P_1'. Since neither D_1 nor Z_1 occurs in P_2' the defender can only respond by $Y_1 \stackrel{d_1}{\to} Z_2 \| W_1$. Let the resulting states be P_1'' and P_2''. Now the attacker performs $Z_2 \stackrel{d_2}{\to} \epsilon$ in P_2'' to which the defender can only respond by $Y_2 \stackrel{d_2}{\to} Z_3 \| W_2$ in P_1'' and so on with Z_j, d_j for $1 \le j \le k$. In the end the defender is forced to perform the transition $Y_k \stackrel{d_k}{\to} Z_{k+1} \| W_k$. Now the action λ is enabled in one process (by the constant Z_{k+1}), but not in the other. Thus the attacker can win and $P_1 \not\sim P_2$. □

Lemma 3. *If Q is not satisfiable then $P_1 \sim P_2$.*

Proof. Let AS (for 'assignments') be the set of subterms containing only constants A_j, B_j, C_j for $1 \le j \le k$. We call a term $t \in AS$ a *faulty assignment* iff there is at least one $m \in \{1, \dots, k\}$ s.t. all three constants A_m, B_m, C_m occur in t. We call the minimal such m the *index* of t, denoted $ind(t)$. Let FAS be the set of faulty assignments. Since Q is not satisfiable, every assignment t that is created by performing one of each pair of actions $x_1/\bar{x}_1 \dots x_n/\bar{x}_n$ is a faulty assignment. Any incomplete assignment t' that is created by an incomplete prefix of choices from $x_1/\bar{x}_1 \dots x_j/\bar{x}_j$ (with $j < n$) must in the end become a faulty assignment once all choices from $x_1/\bar{x}_1 \dots x_n/\bar{x}_n$ have been made. Let $IFAS_j$ be the set of these incomplete faulty assignments created by choosing one of each pair of actions $x_1/\bar{x}_1 \dots x_j/\bar{x}_j$. Let O_j be the set of terms containing only constants Y_l, W_l, Z_l, Z_{l+1} with $l \le j$. To keep the notation simple we define $W_0 := \epsilon$. The symmetric closure of the following relation is a bisimulation.

$$
\begin{array}{l}
\{(X_i \| t \| Z_1^u,\ X_i \| t \| Z_1^v) \mid 1 \le i \le n \ \wedge\ t \in IFAS_{i-1} \ \wedge\ u, v \in \mathbb{N}_0\} \ \cup \\
\{(X_{n+1} \| t \| Z_1^u,\ X_{n+1} \| t \| Z_1^v) \mid t \in FAS \ \wedge\ u, v \in \mathbb{N}_0\} \ \cup \\
\{(Y_1 \| \dots \| Y_k \| t \| Z_1^u,\ Y_1 \| \dots Y_k \| t \| Z_1^v) \mid t \in FAS \ \wedge\ u, v \in \mathbb{N}_0\} \ \cup \\
\{(Y_j \| \dots \| Y_k \| W_{j-1} \| t \| \gamma,\ Y_{j+1} \| \dots \| Y_k \| t \| Z_{j+1} \| W_j \| \gamma') \mid \\
t \in FAS \ \wedge\ j+1 \le ind(t) \ \wedge\ \gamma, \gamma' \in O_{j-1}\} \ \cup \\
\{(Y_j \| \dots \| Y_k \| t \| W_{j-1} \| \gamma,\ Y_{j+1} \| \dots \| Y_k \| t \| W_j \| \gamma') \mid \\
t \in FAS \ \wedge\ j+1 \le ind(t) \ \wedge\ \gamma, \gamma' \in O_{j-1}\} \ \cup \\
\{(Y_{j+1} \| \dots \| Y_k \| t \| Z_{j+1} \| W_j \| \gamma,\ Y_{j+1} \| \dots \| Y_k \| t \| W_j \| \gamma') \mid \\
t \in FAS \ \wedge\ j+1 \le ind(t) \ \wedge\ \gamma, \gamma' \in O_{j-1}\} \ \cup \\
\{(W_j \| Z_{j+1}^u \| Y_{j+1} \| \dots \| Y_k \| \gamma \| t,\ W_j \| Z_{j+1}^u \| Y_{j+1} \| \dots \| Y_k \| \gamma' \| t) \mid \\
u \in \{0, 1\} \ \wedge\ t \in FAS \ \wedge\ 1 \le j \le k \ \wedge\ \gamma, \gamma' \in O_{j-1}\}
\end{array}
$$

Since $(X_1 \| Z_1, X_1)$ is in this relation, we get $P_1 \sim P_2$. □

Theorem 4. *Strong bisimilarity of BPP is co-$\mathcal{NP}$-hard.*

Proof. Directly from Lemma 2 and Lemma 3 and the $\mathcal{NP}$-completeness of 3-SAT.

Note that both P_1 and P_2 are bounded, i.e., they have only finitely many reachable states. It is easy to see that in general the number of reachable states of P_1/P_2 is exponential in the size of the description of Δ. Moreover, the number of reachable states of P_1/P_2 is even exponential up to strong bisimilarity, i.e., they generally have an exponential number of non-bisimilar reachable states. Let $t \in FAS$. Analogously to the definition of the index of t we define $ind'(t)$ as the *maximal* m s.t. all three A_m, B_m, C_m appear in t. Consider the reachable states $X_{n+1}\|t_1\|t_2$, where $t_1\|t_2 \in FAS$ encodes a faulty assignment and the constants A_j, B_j, C_j in t_1 have $j \leq ind'(t_1\|t_2)$ and the constants A_j, B_j, C_j in t_2 have $j > ind'(t_1\|t_2)$. In particular $ind'(t_1\|t_2) = ind'(t_1)$. While the particular structure of t_1 does not matter for bisimilarity (as long as $t_1 \in FAS$), the structure of t_2 does. We have $X_{n+1}\|t_1\|t_2 \not\sim X_{n+1}\|t_1\|t_2'$ for every $t_2' \neq t_2$. Since there are in general exponentially many different such t_2' it follows that P_1/P_2 is at least exponential w.r.t. strong bisimilarity. Thus, our construction does not yield a lower bound for the problem of strong bisimilarity of a BPP and a finite-state process (with polynomially many states). It seems to be impossible to prove a lower bound for this asymmetric problem, since whenever one encodes a sufficiently complex problem (e.g. SAT) into a BPP, this BPP is never bisimilar (neither strongly nor weakly) to any finite-state system of polynomial size (although it can be bisimilar to a finite-state system of exponential size). Thus, we conjecture that strong and weak bisimilarity of a BPP and a finite-state system is decidable in polynomial time. (Is is known that strong and weak bisimilarity of a *normed* BPP and a finite-state system is polynomial [18]).

Now we consider the strong finiteness problem.

<u>STRONG FINITENESS OF BPP</u>

Instance: A BPP process P.

Question: Does there exist a finite-state system F s.t. $P \sim F$?

Finiteness w.r.t. strong bisimilarity is decidable even for general Petri nets [14], and this result carries over immediately to communication-free nets (i.e., BPP). However, the algorithm in [14] consists of two semidecision procedures and gives no upper bound on the complexity. For general Petri nets one gets an $EXPSPACE$ lower bound by reducing the problem if a given place can ever become marked to the finiteness problem. For general Petri nets the problem if a given place can ever become marked is $EXPSPACE$-hard [19, 28]. For communication-free nets (i.e., BPP) this is different. While the reachability problem is $\mathcal{NP}$-complete for communication-free nets [8], it is easy to see that the problem if a given place can ever become marked in a communication-free net is polynomial. Thus, one does not obtain a lower bound for the strong finiteness problem of BPP that way.

It is clear that a *constructive* solution to the problem, i.e., constructing the finite-state system F if it exists, must require at least exponential time. This is because there are BPPs s.t. the smallest finite-state system F that is bisimilar to them has an exponential number of states (in the size of the description of the BPP). However, it is not immediately clear if a simple yes/no answer to the strong finiteness problem must be as hard. The following theorem shows this.

Theorem 5. *Strong finiteness of BPP is co-$\mathcal{NP}$-hard.*

Proof. By a polynomial reduction of 3-SAT to strong infiniteness. Let the formula Q and the set of rules Δ be defined as before and let $\Delta' := \Delta \cup \{X_{n+1} \stackrel{p}{\to} X_{n+1}\|Z_1\}$. We show that the process X_1 w.r.t. the set of rules Δ', denoted (X_1, Δ'), is infinite w.r.t. strong bisimilarity iff Q is satisfiable.

$\Leftarrow$ If Q is satisfiable then there are infinitely many reachable states $Y_1\|\dots\|Y_k\|\gamma\|Z_1^m$ for every $m \in \mathbb{N}_0$, where γ is a term that encodes a satisfying assignment of Q. This means that γ is a parallel composition of constants A_j, B_j, C_j where for every $j \in \{1, \dots, k\}$ at least one of the constants A_j, B_j, C_j does not occur in γ. However, for every $m_1 \neq m_2$ we have $Y_1\|\dots\|Y_k\|\gamma\|Z_1^{m_1} \not\sim Y_1\|\dots\|Y_k\|\gamma\|Z_1^{m_2}$, because the attacker has a winning strategy similar to the one in Lemma 2. Thus (X_1, Δ') is infinite w.r.t. strong bisimilarity.

$\Rightarrow$ Let $\Delta'' := \Delta \cup \{X_{n+1} \stackrel{p}{\to} X_{n+1}\}$. The process (X_1, Δ'') has finitely many reachable states. (However, (X_1, Δ'') has an exponential (in the size of Δ'') number of non-bisimilar reachable states.) If Q is not satisfiable then $(X_1, \Delta') \sim (X_1, \Delta'')$ and is thus finite w.r.t. bisimilarity. The bisimulation relation is the same as in Lemma 3. $\square$

The previous construction shows that the problem if a place can be unmasked (i.e., made to count w.r.t. bisimulation) is $\mathcal{NP}$-hard. Here this particular place was Z_1.

4 Hardness of Weak Bisimilarity for BPP

WEAK BISIMILARITY OF BPP

Instance: Two BPP processes P_1 and P_2.
Question: $P_1 \approx P_2$?

It is still an open question if this problem is decidable. It has been shown to be semidecidable in [8], using the facts that weak bisimulation equivalence on BPPs is semilinear (since it is a congruence on a finitely generated commutative semigroup) and that it is decidable if a given semilinear relation on a BPP is a weak bisimulation. An $\mathcal{NP}$ lower bound for this problem has been shown in [31] (by reduction of a variant of the bin-packing problem), and the co-$\mathcal{NP}$ lower bound of Theorem 4 carries over immediately to weak bisimilarity. Here we prove a Π_2^p-lower bound (in the polynomial hierarchy) that subsumes these results.

Let $Q := Q_1 \wedge \dots \wedge Q_k$ be a boolean formula in 3-CNF over the boolean variables $x_1, \dots, x_n, y_1, \dots, y_n$ with k clauses. We construct BPP processes P_1, P_2 s.t. $P_1 \approx P_2$ iff $\forall(x_1, \dots, x_n)\exists(y_1, \dots, y_n)\, Q$. Since this problem is Π_2^p-complete, we get a Π_2^p-lower bound for the problem of weak bisimilarity.

Let α_i be a parallel composition of constants in $\{Q_1, \dots, Q_k\}$ s.t. constant Q_j appears in α_i iff x_i makes clause Q_j true (i.e., x_i appears positively in Q_j). Let β_i be a parallel composition of constants in $\{Q_1, \dots, Q_k\}$ s.t. constant Q_j appears in β_i iff $\bar{x}_i$ makes clause Q_j true (i.e., x_i appears negatively in Q_j). Let γ_i be a parallel composition of constants in $\{Q_1, \dots, Q_k\}$ s.t. constant Q_j appears in γ_i iff y_i makes clause Q_j true. Let δ_i be a parallel composition of constants in $\{Q_1, \dots, Q_k\}$ s.t. constant Q_j appears in δ_i iff $\bar{y}_i$ makes clause Q_j true. The set of transition rules Δ is defined by

$$
\begin{array}{lll}
X_i & \stackrel{x_i}{\to} X_{i+1} \| \alpha_i & \text{for } 1 \leq i \leq n \\
X_i & \stackrel{\bar{x}_i}{\to} X_{i+1} \| \beta_i & \text{for } 1 \leq i \leq n \\
X'_i & \stackrel{x_i}{\to} X'_{i+1} \| \alpha_i & \text{for } 1 \leq i \leq n \\
X'_i & \stackrel{\bar{x}_i}{\to} X'_{i+1} \| \beta_i & \text{for } 1 \leq i \leq n \\
X_{n+1} & \stackrel{a}{\to} Y_1 \| \ldots \| Y_n & \\
X'_{n+1} & \stackrel{a}{\to} Y_1 \| \ldots \| Y_n & \\
X'_{n+1} & \stackrel{a}{\to} Z & \\
Y_i & \stackrel{\tau}{\to} \gamma_i & \text{for } 1 \leq i \leq n \\
Y_i & \stackrel{\tau}{\to} \delta_i & \text{for } 1 \leq i \leq n \\
Q_j & \stackrel{q_j}{\to} Q_j & \text{for } 1 \leq j \leq k \\
Z & \stackrel{q_j}{\to} Z & \text{for } 1 \leq j \leq k
\end{array}
$$

Let $P_1 := (X_1, \Delta)$ and $P_2 := (X'_1, \Delta)$.

Lemma 6. *If* $\forall(x_1, \ldots, x_n)\exists(y_1, \ldots, y_n)\, Q$ *is false then* $P_1 \not\approx P_2$.
Proof. If $\forall(x_1, \ldots, x_n)\exists(y_1, \ldots, y_n)\, Q$ is false then $\exists(x_1, \ldots, x_n)\forall(y_1, \ldots, y_n)\neg Q$. The attacker chooses these values for $x_1, \ldots, x_n$ by choosing $x_i/\bar{x}_i$. The defender can only copy these moves. Then the attacker chooses the transition $X'_{n+1} \stackrel{a}{\to} Z$. The defender can only respond by $X_{n+1} \stackrel{a}{\to} Y_1 \| \ldots \| Y_n$ and then a sequence of silent τ-actions ending in a state t. By definition of Δ and since $\exists(x_1, \ldots, x_n)\forall(y_1, \ldots, y_n)\neg Q$ there will be at least one action q_j (with $1 \leq j \leq k$) that is not enabled by t (and cannot made to be enabled by τ-moves). However, all q_j are enabled by Z. Thus, the attacker has a winning strategy and $P_1 \not\approx P_2$. □

Lemma 7. *If* $\forall(x_1, \ldots, x_n)\exists(y_1, \ldots, y_n)\, Q$ *then* $P_1 \approx P_2$.
Proof. The attacker can choose the assignment for $x_1, \ldots, x_n$. The defender can only imitate these choices. If the attacker chooses the transition $X_{n+1} \stackrel{a}{\to} Y_1 \| \ldots \| Y_n$ or $X'_{n+1} \stackrel{a}{\to} Y_1 \| \ldots \| Y_n$ then the defender can respond in such a way that the two processes become equal and the defender wins. If the attacker chooses $X'_{n+1} \stackrel{a}{\to} Z$ then the defender can (by a long internal move of τ-actions) choose the values for $y_1, \ldots, y_n$ on his side. Since $\forall(x_1, \ldots, x_n)\exists(y_1, \ldots, y_n)\, Q$ there are choices for $y_1, \ldots, y_n$ s.t. in the resulting state all actions $q_1, \ldots, q_k$ are permanently enabled. Since $q_1, \ldots, q_k$ are also permanently enabled by Z in the other process and all other actions are not, the defender wins. Thus, the defender has a winning strategy and $P_1 \approx P_2$. □

Theorem 8. *Weak bisimilarity of BPP is* Π_2^p*-hard.*
Proof. Directly from Lemma 6 and Lemma 7.

<u>WEAK FINITENESS OF BPP</u>
Instance: A BPP process P.
Question: Does there exist a finite-state system F s.t. $P \approx F$?

We show that the weak finiteness problem for BPP is also Π_2^p-hard by using the previously defined processes P_1 and P_2 and constructing a new process P that is weakly finite iff $P_1 \approx P_2$. Let Δ' be $\Delta \cup \Gamma$, where Γ is the following set of transition rules:

$I \xrightarrow{\tau} I\|C \quad I \xrightarrow{\tau} \epsilon \quad C \xrightarrow{c} \epsilon \quad D \xrightarrow{c} E \quad D \xrightarrow{c} E' \quad D \xrightarrow{e} X_1\|S$

$D \xrightarrow{e} X_1'\|S \quad E \xrightarrow{c} E \quad E' \xrightarrow{c} E' \quad E \xrightarrow{e} X_1\|S \quad E' \xrightarrow{e} X_1'\|S \quad S \xrightarrow{c} S$

Let $P := (I\|D, \Delta')$.

Lemma 9. *If $P_1 \not\approx P_2$ then P is not weakly finite.*

Proof. P has infinitely many non-weakly-bisimilar states $D\|C^i$ for all $i \in \mathbb{N}$. It suffices to show that $D\|C^j \not\approx D\|C^i$ for $j > i$. The attacker has the following winning strategy. He does action c exactly $i+1$ times in $D\|C^j$ and reaches the state $D\|C^{j-i-1}$. The defender can respond in different ways in $D\|C^i$, but the reached state will always be either $E\|C^k$ or $E'\|C^k$ for some $k \leq i$. In the first case the attacker does the transition $D \xrightarrow{e} X_1'\|S$. The defender can only respond by $E \xrightarrow{e} X_1\|S$ and the new state in the bisimulation game is $(X_1'\|S\|C^{j-i-1}, X_1\|S\|C^k)$. This is not weakly bisimilar, because $P_1 \not\approx P_2$. The second case is symmetric with X_1 and X_1' exchanged. □

Lemma 10. *If $P_1 \approx P_2$ then P is weakly finite.*

Proof. Let Γ' be Γ where X_1' is replaced by X_1 and $\Delta'' := \Delta \cup \Gamma'$. Since $P_1 \approx P_2$ and weak bisimilarity is a congruence on BPP, we get $P = (I\|D, \Delta') \approx (I\|D, \Delta'')$. It is easy to see that $(I\|D, \Delta'') \approx (E, \Delta'')$, because $S \xrightarrow{c} S$. Thus $P \approx (E, \Delta'')$. However, (E, Δ'') has only finitely many reachable states. □

Theorem 11. *Weak finiteness of BPP is Π_2^p-hard.*

Proof. By Lemmas 6, 7, 9 and 10. □

5 Conclusion

The following table summarizes known results about the complexity of bisimulation problems for several classes of infinite-state systems. New results are in boldface.

	$\sim F$	$\sim$	$\approx F$	$\approx$
FS	$\mathcal{P}$ [2, 27]	$\mathcal{P}$ [2, 27]	$\mathcal{P}$ [2, 27]	$\mathcal{P}$ [2, 27]
BPA	$\mathcal{P}$ [18]	$\in 2-EXPTIME$ [3]	$\mathcal{P}$ [18]	$PSPACE$-hard [31]
PDA	$\in EXPTIME$ [15] $PSPACE$-hard [23]	decidable [29] $PSPACE$-hard [23]	$\in EXPTIME$ [15] $PSPACE$-hard [23]	$PSPACE$-hard [31]
BPP	$\in PSPACE$ [15]	decidable [7] **co-NP-hard**	$\in PSPACE$ [15]	$\mathbf{\Pi_2^p}$**-hard**
PA	decidable [15]	**co-NP-hard**	decidable [15]	$PSPACE$-hard [31]
PAD	decidable [15] $PSPACE$-hard [23]	$PSPACE$-hard [23]	decidable [15] $PSPACE$-hard [23]	$PSPACE$-hard [31]
PN	decidable [16, 15] $EXPSPACE$-hard	undecidable [13]	undecidable [13]	undecidable [13]
PAN	$EXPSPACE$-hard	undecidable [13]	undecidable [13]	undecidable [13]
PRS	$EXPSPACE$-hard	undecidable [13]	undecidable [13]	undecidable [13]

The different columns in the table above show the results about the following problems: strong bisimilarity with finite automata, strong bisimilarity of two infinite-state systems, weak bisimilarity with finite automata and weak bisimilarity of two infinite-state systems.

The following table summarizes results about the problems of strong and weak finiteness. New results are in boldface.

	strong finiteness	weak finiteness
BPA	$\in 2-EXPTIME$ [4, 3]	?
PDA	$PSPACE$-hard [23]	$PSPACE$-hard [23]
BPP	decidable [14] **co-NP-hard**	$\mathbf{\Pi_2^p}$**-hard**
PA	**co-NP-hard**	$\mathbf{\Pi_2^p}$**-hard**
PAD	$PSPACE$-hard [23]	$PSPACE$-hard [23]
PN	decidable [14] $EXPSPACE$-hard	undecidable [14]
PAN/PRS	$EXPSPACE$-hard	undecidable [14]

Some more results are known about the restricted subclasses of these systems that satisfy the 'normedness condition' (e.g. [12, 11, 10, 17, 18]).

References

[1] J.C.M. Baeten and W.P. Weijland. Process algebra. *Cambridge Tracts in Theoretical Computer Science*, 18, 1990.

[2] J. Balcazar, J. Gabarro, and M. Santha. Deciding bisimilarity is P-complete. *Formal Aspects of Computing*, 4:638–648, 1992.

[3] O. Burkart, D. Caucal, and B. Steffen. An elementary bisimulation decision procedure for arbitrary context-free processes. In *MFCS'95*, volume 969 of *LNCS*. Springer Verlag, 1995.

[4] O. Burkart, D. Caucal, and B. Steffen. Bisimulation collapse and the process taxonomy. In U. Montanari and V. Sassone, editors, *Proceedings of CONCUR'96*, volume 1119 of *LNCS*. Springer Verlag, 1996.

[5] D. Caucal. On the regular structure of prefix rewriting. *Journal of Theoretical Computer Science*, 106:61–86, 1992.

[6] S. Christensen. *Decidability and Decomposition in Process Algebras*. PhD thesis, Edinburgh University, 1993.

[7] S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation equivalence is decidable for Basic Parallel Processes. In E. Best, editor, *Proceedings of CONCUR 93*, volume 715 of *LNCS*. Springer Verlag, 1993.

[8] J. Esparza. Petri nets, commutative context-free grammars and Basic Parallel Processes. In Horst Reichel, editor, *Fundamentals of Computation Theory*, volume 965 of *LNCS*. Springer Verlag, 1995.

[9] J. Esparza and A. Kiehn. On the model checking problem for branching time logics and Basic Parallel Processes. In *CAV'95*, volume 939 of *LNCS*, pages 353–366. Springer Verlag, 1995.

[10] Y. Hirshfeld and M. Jerrum. Bisimulation equivalence is decidable for normed process algebra. In *Proc. of ICALP'99*, volume 1644 of *LNCS*. Springer Verlag, 1999.

[11] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theoretical Computer Science*, 158:143–159, 1996.

[12] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial-time algorithm for deciding bisimulation equivalence of normed Basic Parallel Processes. *Journal of Mathematical Structures in Computer Science*, 6:251–259, 1996.

[13] P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148:281–301, 1995.

[14] P. Jančar and J. Esparza. Deciding finiteness of Petri nets up to bisimulation. In F. Meyer auf der Heide and B. Monien, editors, *Proceedings of ICALP'96*, volume 1099 of *LNCS*. Springer Verlag, 1996.

[15] P. Jančar, A. Kučera, and R. Mayr. Deciding bisimulation-like equivalences with finite-state processes. In *Proc. of ICALP'98*, volume 1443 of *LNCS*. Springer Verlag, 1998.

[16] P. Jančar and F. Moller. Checking regular properties of Petri nets. In Insup Lee and Scott A. Smolka, editors, *Proceedings of CONCUR'95*, volume 962 of *LNCS*. Springer Verlag, 1995.

[17] A. Kučera. Regularity is decidable for normed PA processes in polynomial time. In *Foundations of Software Technology and Theoretical Computer Science (FST&TCS'96)*, volume 1180 of *LNCS*. Springer Verlag, 1996.

[18] A. Kučera and R. Mayr. Weak bisimilarity with infinite-state systems can be decided in polynomial time. In *Proc. of CONCUR'99*, volume 1664 of *LNCS*. Springer Verlag, 1999.

[19] R. Lipton. The reachability problem requires exponential space. Technical Report 62, Department of Computer Science, Yale University, January 1976.

[20] R. Mayr. Weak bisimulation and model checking for Basic Parallel Processes. In *Foundations of Software Technology and Theoretical Computer Science (FST&TCS'96)*, volume 1180 of *LNCS*. Springer Verlag, 1996.

[21] R. Mayr. Tableau methods for PA-processes. In D. Galmiche, editor, *Analytic Tableaux and Related Methods (TABLEAUX'97)*, volume 1227 of *LNAI*. Springer Verlag, 1997.

[22] R. Mayr. *Decidability and Complexity of Model Checking Problems for Infinite-State Systems*. PhD thesis, TU-München, 1998.

[23] R. Mayr. On the complexity of bisimulation problems for pushdown automata. 2000.

[24] R. Mayr. Process rewrite systems. *Information and Computation*, 156(1):264–286, 2000.

[25] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[26] F. Moller. Infinite results. In Ugo Montanari and Vladimiro Sassone, editors, *Proceedings of CONCUR'96*, volume 1119 of *LNCS*. Springer Verlag, 1996.

[27] R. Paige and R. Tarjan. Three partition refinement algorithms. *SIAM Journal of Computing*, 16(6):973–989, 1987.

[28] J.L. Peterson. *Petri net theory and the modeling of systems*. Prentice-Hall, 1981.

[29] G. Sénizergues. Decidability of bisimulation equivalence for equational graphs of finite out-degree. In *Proc. of FOCS'98*. IEEE, 1998.

[30] C. Stirling. The joys of bisimulation. In *Proc. of MFCS'98*, volume 1450 of *LNCS*, pages 142–151. Springer Verlag, 1998.

[31] J. Stříbrná. Hardness results for weak bisimilarity of simple process algebras. *Electronic Notes in Theoretical Computer Science (ENTCS)*, 18, 1998.

[32] W. Thomas. On the Ehrenfeucht-Fraïssé game in theoretical computer science. In *Proc. of TAPSOFT'93*, volume 668 of *LNCS*, pages 559–568. Springer Verlag, 1993.