Weak Bisimulation and Model Checking for Basic Parallel Processes

Richard Mayr

Institut für Informatik, Technische Universität München, Arcisstr. 21, D-80290 München, Germany; e-mail: mayrri@informatik.tu-muenchen.de

Abstract. Basic Parallel Processes (BPP) are a natural subclass of CCS infinite-state processes. They are also equivalent to a special class of Petri nets. We show that unlike for general Petri nets, it is decidable if a BPP and a finite-state system are weakly bisimilar. To the best of our knowledge, this is the first decidability result for weak bisimulation and a non-trivial class of infinite-state systems. We also show that the model checking problem for BPPs and the branching time logic UB^- is PSPACE-complete. This settles a conjecture of [4].

Keywords: Basic Parallel Processes, bisimulation, model checking.

1 Introduction

Bisimulation equivalence [10], has become one of the most successful equivalence notions in concurrency theory, both with respect to theoretical research and to applications. In recent years, the decidability of bisimulation equivalence on several classes of infinite-state systems has been intensely studied, and a number of positive results have been obtained. Strong bisimilarity between

- two context-free processes,
- two Basic Parallel Processes (BPPs),
- a Petri net and a finite-state system

has been shown to be decidable [3, 2, 8].

A natural next step in this line of research is to extend at least some of these encouraging results to weak bisimilarity, an equivalence that allows to ignore silent internal actions. However, this task has proved to be difficult, and to the best of our knowledge only undecidability results have been obtained so far (i.e. weak bisimilarity of a Petri net and a finite-state system is undecidable [7]). In this paper we report on a first positive result: weak bisimilarity between a BPP and a finite-state system is decidable.

In the second part of the paper we shift our attention to the model checking problem for BPPs, called just the model checking problem in the sequel. This is the problem of deciding if a given BPP satisfies a property coded as a formula in a certain temporal logic. It has been shown in [4] that the model checking problem is undecidable for formulae built out of boolean operators, EX (for some successor) and EG (for some path always in the future), but decidable when the temporal operators are EX and EF (for some path eventually in the future). Therefore, the logic with the latter set of operators, called UB^- in [4], seems to be the largest branching time logic with a decidable model checking problem. It was shown in [4] that the model checking problem for UB^- is PSPACE-hard even for finite-state BPPs (notice that the size of the problem is the size of the formula plus the size of the BPP, and not the size of its associated transition system), but the exact complexity of the problem remained open. We prove that this problem is PSPACE-complete by showing that it only requires polynomial space, even for infinite-state BPPs.

2 Preliminaries

Definition 1. A binary relation R over the states of a labelled transition system (for short: LTS) is a *strong bisimulation* (often simply called bisimulation) iff

$$\begin{aligned}\forall (s_1, s_2) \in R\ \forall a \in Act.\ &(s_1 \xrightarrow{a} s_1' \ \Rightarrow \exists s_2 \xrightarrow{a} s_2'.\ s_1' R s_2') \ \wedge \\ &(s_2 \xrightarrow{a} s_2' \ \Rightarrow \exists s_1 \xrightarrow{a} s_1'.\ s_1' R s_2')\end{aligned}$$

Two states s_1 and s_2 are *strongly bisimilar* iff there is a strong bisimulation R such that $s_1 R s_2$. This definition can be extended to states in different transition systems by putting them 'side by side' and considering them as a single transition system. It is easy to see that there always exists a largest strong bisimulation which is an equivalence relation called **strong bisimulation equivalence**. It is denoted by $\sim$.

Strong bisimulation equivalence is sometimes too strict. Processes can contain silent internal actions (labelled by τ) which should not be externally visible. Therefore another equivalence called weak bisimulation equivalence is defined that treats these τ-actions accordingly.

Definition 2. Let $\stackrel{a}{\Rightarrow} := (\xrightarrow{\tau})^* \xrightarrow{a} (\xrightarrow{\tau})^*$ for $a \in Act$ and $\stackrel{\hat{a}}{\Rightarrow} := \begin{cases} \stackrel{a}{\Rightarrow}, & \text{if } a \neq \tau \\ (\xrightarrow{\tau})^*, & \text{if } a = \tau \end{cases}$
A binary relation R over the states of an LTS is a *weak bisimulation* if

$$\begin{aligned}\forall (s_1, s_2) \in R\ \forall a \in Act.\ &(s_1 \xrightarrow{a} s_1' \ \Rightarrow \exists s_2 \stackrel{\hat{a}}{\Rightarrow} s_2'.\ s_1' R s_2') \ \wedge \\ &(s_2 \xrightarrow{a} s_2' \ \Rightarrow \exists s_1 \stackrel{\hat{a}}{\Rightarrow} s_1'.\ s_1' R s_2')\end{aligned}$$

Two states s_1 and s_2 are *weakly bisimilar* iff there is a weak bisimulation R such that $s_1 R s_2$. Again this can be extended to states in different transition systems. There always exists a largest weak bisimulation which is an equivalence relation called **weak bisimulation equivalence**. It is denoted by $\approx$. It is clear that $\sim \subseteq \approx$ for every LTS.

A process that is weakly bisimilar to a finite-state LTS is called weakly finite.

Another approach to bisimulation is via bisimulation games. A game consists of a triple $(\mathcal{T}, s_1, s_2)$, where $\mathcal{T}$ is an LTS and s_1, s_2 are states, and two players A (Attacker) and D (Defender). It is the aim of player A to prove that s_1 and s_2 are not bisimilar, while player D attempts to frustrate this. A round in the game goes like this:

1. Player A chooses an $i \in \{1,2\}$, an action a and a move $s_i \xrightarrow{a} s_i'$. If it is impossible to make any move in s_1 or s_2 then player D wins.
2. Then player D takes the $j \in \{1,2\}$ s.t. $i \neq j$ and tries to imitate the move, i.e. find a s_j' s.t. $s_j \xrightarrow{a} s_j'$. If this is impossible then player A wins.

After this round the resulting state is $(\mathcal{T}, s_1', s_2')$ and the play repeats. Player D wins every infinite game.

s_1 and s_2 are *strongly bisimilar up to n* if there is a strategy for player D to defend (prevent player A from winning) for at least n rounds. This is denoted as $s_1 \sim_n s_2$. It is easy to see that $\sim_n$ is an equivalence relation for every $n \in \mathbb{N}$. If the graph of $\mathcal{T}$ is finitely branching, then $\sim = \bigcap_{n=1}^{\infty} \sim_n$.

Weak bisimulation up to n (denoted by $\approx_n$) is defined in the same way, except that player D makes moves $s_j \stackrel{\hat{a}}{\Rightarrow} s_j'$ instead of $s_j \xrightarrow{a} s_j'$. It follows immediately that $\sim_n \subset \approx_n$. Unlike for strong bisimulation the relation $\approx_n$ is not an equivalence relation in general. Also $\approx \neq \bigcap_{n=1}^{\infty} \approx_n$, because $\bigcap_{n=1}^{\infty} \approx_n \not\subset \approx$. The other inclusion $\approx \subset \bigcap_{n=1}^{\infty} \approx_n$ holds.

The Basic Parallel Processes (BPP) were introduced by Christensen in his Ph.D. dissertation [1] as a very natural subclass of the class of CCS processes [10]. They are a simple model of infinite-state concurrent systems. Assume a countably infinite set of atomic actions $Act = \{a, b, c, \ldots\}$ and a countably infinite set of process variables $Var = \{X, Y, Z, \ldots\}$. The class of BPP expressions is defined by the following abstract syntax [1, 2]:

$$E \stackrel{\text{def.}}{=} 0 \mid X \mid aE \mid E + E \mid E\|E$$

$\alpha, \beta, \gamma, \ldots$ denote merges of process variables. A BPP is defined by a family of recursive equations $\{X_i := E_i \mid 1 \leq i \leq n\}$ where the X_i are distinct and the E_i are BPP expressions at most containing the variables $\{X_1, \ldots, X_n\}$. It will be assumed that every variable occurrence in the E_i is *guarded*, i.e. appears within the scope of an action prefix. The variable X_1 is singled out as the *leading variable* and $X_1 := E_1$ is called the *leading equation*. Any finite family of BPP equations determines a labelled transition system. For every $a \in Act$ the transition relation $\xrightarrow{a}$ is the least relation satisfying the following inference rules:

$$aE \xrightarrow{a} E \quad \frac{E \xrightarrow{a} E'}{E+F \xrightarrow{a} E'} \quad \frac{F \xrightarrow{a} F'}{E+F \xrightarrow{a} F'} \quad \frac{E \xrightarrow{a} E'}{E\|F \xrightarrow{a} E'\|F} \quad \frac{F \xrightarrow{a} F'}{E\|F \xrightarrow{a} E\|F'} \quad \frac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'}(X{:=}E)$$

BPP processes generate finitely branching transition graphs, i.e. $\{F \mid E \xrightarrow{a} F\}$ is finite for each E and a. This would not be true if unguarded expressions were allowed. For example, the process $X := a + a\|X$ generates an infinitely branching transition graph. A BPP is in *normal form*, if every expression E_i at the right hand side of an equation is of the form $a_1\alpha_1 + \ldots + a_n\alpha_n$. It is shown in [1] that every BPP is strongly bisimilar to a BPP in normal form.

A BPP in normal form can be translated into a labelled Petri net s.t. their transition graphs are isomorphic. Introduce a place for each process variable and a transition for each transition rule. For a rule $X \xrightarrow{a} Y_1^{m_1} \cdots Y_n^{m_n}$ introduce a transition t labelled by a, an arc labelled by 1 leading from place X to t and arcs labelled by m_i leading from t to places Y_i. It is important to note that in these nets every transition has exactly one input place with an arc labelled by 1. Such Petri nets will be called *communication-free nets*. Communication-free nets will be denoted by N and their markings by Σ. Translating a communication-free net N back into a BPP-algebra is analogous. Any marking Σ of N can then be translated into a BPP in this algebra. So there is a one-to-one correspondence between BPP-algebras and communication-free nets, as well as between elements of the algebra and markings of the net.

Definition 3. For labelled Petri nets there is a labeling function $L : T \rightarrow Act$ that assigns actions to the transitions. The labeling function L is extended to sequences of transitions in the standard way. If a sequence of transitions σ is fireable at a marking Σ and leads to a new marking Σ' this is denoted by $\Sigma \xrightarrow{\sigma} \Sigma'$.

A sequence σ' is called a **smaller sequence** than σ if its Parikh-vector is smaller (componentwise) than the one of σ. In this case σ is called a **greater sequence** than σ'. σ' is called a **subsequence** of σ if it is a smaller sequence and the transitions occur in the same order.

Definition 4. The *size* of a communication-free net $N = (S, T, W)$ is the space needed to describe it. This implies that if n is the size of N then $|S| \leq n$, $|T| \leq n$ and $\forall t \in T\ \forall s \in S. W(t, s) \leq 2^n$, where W is the weight function that assigns weights to the arcs in the net. The size of a BPP-algebra is the size of the corresponding communication-free net. The size of a marking Σ of N is defined as the number of tokens in it. So the space needed to describe Σ is $O(log(size(\Sigma)))$.

3 Deciding Weak Bisimilarity of a BPP and a Finite State Labelled Transition System

Strong bisimilarity is undecidable for general Petri nets [7] and CCS [10]. However, it is decidable, if a Petri net and a finite-state LTS are strongly bisimilar [8].

On the other hand it is undecidable if a general Petri net and a finite-state LTS are weakly bisimilar [7]. In this section we show that it is decidable if a BPP and a finite-state LTS are weakly bisimilar.

The basic idea is to show that if a weakly finite BPP can reach a state (via a sequence of τ-transitions) that is weakly bisimilar to a finite-state LTS, then it can reach such a state via a sequence of bounded length.

Lemma 5 Dickson's lemma. *For every infinite sequence of vectors* $M_1, M_2, \ldots$ *in* $\mathbb{N}^k$ *there are* $i < j$ *s.t.* $M_i \leq M_j$ *(*$\leq$ *taken componentwise).*

Lemma 6. *Weak bisimilarity is a congruence for BPP. Let* $\alpha, \beta, \gamma, \delta$ *be BPP-processes.* $\alpha \approx \beta \ \wedge \ \gamma \approx \delta \ \Rightarrow \ \alpha\gamma \approx \beta\delta$*.*

Proof The relation $\{(xz, yw) \mid x \approx y, z \approx w\}$ is a weak bisimulation containing $(\alpha\gamma, \beta\delta)$. This is because BPPs are equivalent to communication-free nets, where every transition has only one place in its preset with an arc labelled by 1. $\square$

Lemma 7. *Let* R *be an LTS with* m *states. Let* $r_1, r_2 \in R$ *be states of* R*. Then* $r_1 \approx r_2 \ \Leftrightarrow \ r_1 \approx_{(m^2-m)/2} r_2$*.*

Proof As R has a finite number of states the relation $\stackrel{\hat{a}}{\Rightarrow}$ is finitely branching for any $a \in Act$. Therefore $r_1 \approx r_2 \Leftrightarrow \forall n \in \mathbb{N}. r_1 \approx_n r_2$. So if $r_1 \not\approx r_2$, then there must be an $i \in \mathbb{N}$ and a strategy for player A, s.t. he can win every bisimulation game in $\leq i$ rounds. A bisimulation game is described by a sequence of pairs $(r, r') \in R \times R$. There are at most $(m^2 - m)/2$ different such pairs (r, r') (modulo symmetry) with $r \neq r'$. So there must be such an i with $i \leq (m^2 - m)/2$, and therefore $r_1 \not\approx r_2 \Rightarrow r_1 \not\approx_{(m^2-m)/2} r_2$. The other direction is trivial. $\square$

Definition 8. In a communication-free net tokens can move freely through the net, because every transition has only one place in its preset and the arc leading from this place to the transition is labelled by 1. When a transition fires it takes a token from the place in its preset and puts some tokens on the places in its postset. If there are places in the postset of the transition one can freely choose any one of these tokens and call it the *continuation* of the original token. All the others will be called *spin-offs*. Assume a firing sequence σ. A subsequence σ' of σ that does nothing but move a token back to its original place, generating some spin-offs on the way is called a *cycle*. When a cycle is possible it can be repeated an arbitrary number of times, because the resulting marking is bigger than the original one. A cycle does not change a marking, except that it generates some new tokens (the spin-offs).

Sequences not containing any cycles have bounded length.

Lemma 9. *Let* (N, Σ) *be a communication-free net with marking* Σ*. Let* n *be the size of* N *and* x *the number of tokens in* Σ*. Let* σ *be a firing sequence starting in* Σ *and not containing any cycle* σ' *as subsequence. Then* $length(\sigma) \leq x\frac{2^{n^2-n}-1}{2^n-1}$*.*

Proof Any non-cyclic path in N has a length of at most $n - 1$. As there are no cycles a token can move at most $n - 1$ steps. The firing of a transition increases the number of tokens in the net by at most $2^n - 1$. So the sequence σ has a maximal length of $\sum_{i=0}^{n-2} x * (2^n)^i \ = \ x\frac{1-(2^n)^{n-1}}{1-2^n} = x\frac{2^{n^2-n}-1}{2^n-1}$ $\square$

Definition 10. Let N be a Petri net and S the set of its places. For every $x \in \mathbb{N}$ the relation $\leq_x$ on the set of markings of N is defined by

$$\begin{aligned} \Sigma \leq_x \Sigma' :\Leftrightarrow \ & \Sigma \leq \Sigma' \quad \wedge \\ & \forall s \in S.\ \Sigma(s) < \Sigma'(s) \Rightarrow \Sigma(s) \geq x \end{aligned}$$

For every x the relation $\leq_x$ is a partial order.

The following lemmas are used to show that two weakly finite markings Σ, Σ' of a communication-free net with $\Sigma \leq_k \Sigma'$ are weakly bisimilar if k is sufficiently large.

Lemma 11. *Let N be a communication-free net with n places and two markings Σ_1 and Σ_2 s.t. $\Sigma_1 \leq_x \Sigma_2$ for an $x \geq n$. Then for any sequence $\Sigma_2 \xrightarrow{\sigma} \Sigma_2'$ s.t. $L(\sigma) \in \tau^* a \tau^*$ there is a subsequence σ' s.t. $L(\sigma') \in \tau^* a \tau^*$, $\Sigma_1 \xrightarrow{\sigma'} \Sigma_1'$ and $\Sigma_1' \leq_{[x/n]} \Sigma_2'$.*

Proof The only difference between σ and the subsequence σ' are the moves of tokens starting in places s s.t. $\Sigma_1(s) < \Sigma_2(s)$. For such places $\Sigma_1(s) \geq x$. By a sequence of τ-moves in σ a token starting in s is either moved to a place s' (possibly doing cycles on the way, and generating spin-offs) or finally vanishes (by a transition that has no places in its postset). In σ' some, but not necessarily all (at most $[x/n]$) of these moves are imitated. As $\Sigma_1(s) \geq x$ there are enough tokens to move up to $[x/n]$ tokens to any place s' reachable from s, if necessary. If $\Sigma_2'(s) \geq [x/n]$ then keep $[x/n]$ tokens on s, otherwise move further tokens away, s.t. $\Sigma_2'(s) = \Sigma_1'(s)$. All the cycles in σ are imitated in σ'. This is no problem, since cycles just generate new tokens. The single a-move is also imitated. So one gets a Σ_1' that on any place either has the same number of tokens as Σ_2', or at least $[x/n]$. Thus $\Sigma_1' \leq_{[x/n]} \Sigma_2'$. □

Lemma 12. *Let N be a communication-free net with n places and two markings Σ_1 and Σ_2 s.t. $\Sigma_1 \leq_x \Sigma_2$ for an $x \geq n$. Then for any sequence $\Sigma_1 \xrightarrow{\sigma} \Sigma_1'$ s.t. $L(\sigma) \in \tau^* a \tau^*$ there is a greater sequence σ' s.t. $L(\sigma') \in \tau^* a \tau^*$, $\Sigma_2 \xrightarrow{\sigma'} \Sigma_2'$ and $\Sigma_1' \leq_{[x/n]} \Sigma_2'$.*

Proof Again the only difference between σ and σ' are the moves of tokens starting in places s s.t. $\Sigma_1(s) < \Sigma_2(s)$. For such places $\Sigma_1(s) \geq x$. By a sequence of τ-moves in σ a token starting in s is either moved to a place s' (possibly doing cycles on the way, and generating spin-offs) or finally vanishes (by a transition which has no places in its postset). As $\Sigma_1(s) \geq x$ at least one of the following conditions must be satisfied:

1. There is a path in the net going only through τ-transitions that leads from s to some place s' such that in σ tokens are moved from s to s' s.t. $\Sigma_1'(s') \geq [x/n]$. (Note that it is possible that $s = s'$).
2. There is a path in the net going only through τ-transitions that have only one place in their postset leading from s to a τ-transition t with $t^{\cdot} = \{\}$. (Tokens on s can vanish by a sequence of τ-moves).

In σ' first imitate all the moves of σ. If $\Sigma_1'(s) < [x/n]$ then do the following, depending on which of the conditions is satisfied.

1. If the first condition is satisfied then track the moves of a token that moves from s to s' in σ. Imitate these moves in σ' for all the extra tokens on place s in Σ_2 until the same number of tokens on s is reached as in Σ_1'. The result may be that $\Sigma_2'(s') \gg \Sigma_1'(s')$, but still $\Sigma_1'(s') \geq [x/n]$.

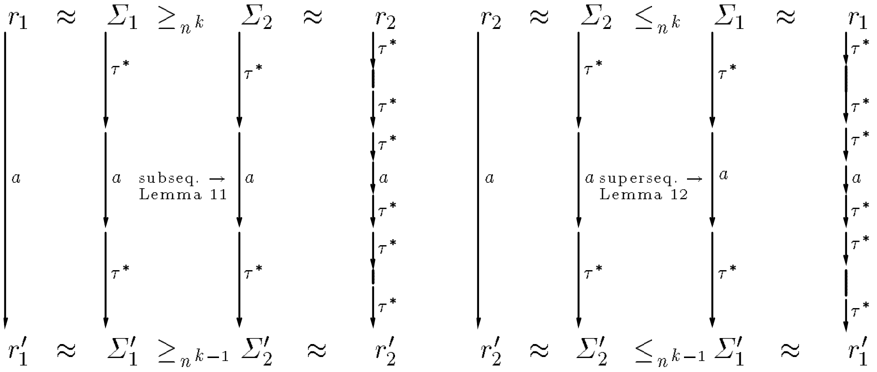
2. In this case let the extra tokens on s in the marking Σ_2 vanish by sequences of τ-moves as described in the second condition, and get $\Sigma_2'(s) = \Sigma_1'(s)$.

Adding these extra τ-moves to σ yields a σ' s.t. σ' is a greater sequence than σ, $\Sigma_2 \xrightarrow{\sigma'} \Sigma_2'$ and $\Sigma_2' \geq_{[x/n]} \Sigma_1'$. $\square$

Lemma 13. *Let Σ_1, Σ_2 be markings of a communication-free net of size n, and r_1, r_2 states in a finite-state LTS. If $r_1 \approx \Sigma_1 \geq_{n^k} \Sigma_2 \approx r_2$ then $r_1 \approx_k r_2$.*

Proof by induction on k. If player A makes a move $r_1 \xrightarrow{a} r_1'$, then there exists $\Sigma_1 \xrightarrow{\sigma} \Sigma_1'$ s.t. $L(\sigma) \in \tau^* a \tau^*$ and $r_1' \approx \Sigma_1'$. By Lemma 11 there is a subsequence σ' s.t. $\Sigma_2 \xrightarrow{\sigma'} \Sigma_2'$, $L(\sigma') \in \tau^* a \tau^*$ and $\Sigma_1' \geq_{n^{k-1}} \Sigma_2'$. As $r_2 \approx \Sigma_2$ there is a sequence σ'' s.t. $r_2 \xrightarrow{\sigma''} r_2'$, $L(\sigma'') \in \tau^* a \tau^*$ and $\Sigma_2' \approx r_2'$. By induction hypothesis the result follows. For an attack $r_2 \xrightarrow{a} r_2'$ the strategy is analogous, except that Lemma 12 is used and σ' is a greater sequence than σ (see Figure 1). $\square$

Fig. 1. Strategy for player D to defend for at least k rounds



The relation $\bigcup_{i=0}^{n} (\xrightarrow{\tau})^i$ is also denoted by $\xrightarrow{\tau \leq n}$. Let

$$A(\Sigma, a, l) := \begin{cases} \{\tilde{\Sigma} \mid \Sigma \xrightarrow{\tau \leq l} \xrightarrow{a} \xrightarrow{\tau \leq l} \tilde{\Sigma}\}, & \text{if } a \neq \tau \\ \{\tilde{\Sigma} \mid \Sigma \xrightarrow{\tau \leq 2l+1} \tilde{\Sigma}\}, & \text{if } a = \tau \end{cases}$$

Now we show a bound for the length of a sequence needed to reach a marking that is weakly bisimilar to a finite-state LTS.

Lemma 14. *Let N be a communication-free net of size n and Σ a marking of N containing x tokens. If $\Sigma \xrightarrow{\sigma} \Sigma'$ then for every $k \in \mathbb{N}$ there is a subsequence $\tilde{\sigma}$ of σ s.t. $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$ and $\tilde{\Sigma} \leq_k \Sigma'$ and*

$$length(\tilde{\sigma}) \leq n^2 k + (x + (2^n - 1) n^2 k) * \frac{2^{n^2 - n} - 1}{2^n - 1}$$

Proof $\tilde{\sigma}$ is the same as σ, except that it possibly contains fewer cycles. What is the maximal number of cycles in $\tilde{\sigma}$ that are needed to reach such a $\tilde{\Sigma}$ with $\tilde{\Sigma} \leq_k \Sigma'$? Cycles just generate new tokens, and at most k new tokens need to be produced per place in N. So at most $n * k$ cycles are needed in $\tilde{\sigma}$. In every cycle $\leq n$ transitions are fired, so $\leq (2^n - 1)n^2k$ new tokens are produced. So at most $x + (2^n - 1)n^2k$ tokens are in the net for moves without cycles. By Lemma 9 at most $(x + (2^n - 1)n^2k) * \frac{2^{n^2-n}-1}{2^n-1}$ non-cyclic moves can be made [1]. By adding the numbers of moves belonging to cycles and the non-cyclic moves we get $length(\tilde{\sigma}) \leq n^2k + (x + (2^n - 1)n^2k) * \frac{2^{n^2-n}-1}{2^n-1}$. □

Lemma 15. *Let N be a communication-free net and Σ a marking of N. Let R be a finite-state system and $r \in R$ a state of R. Let n be the size of the net N, x the number of tokens in Σ and m the number of states in R. There is a function $l : \mathbb{N}^3 \to \mathbb{N}$ s.t. $(\Sigma \approx r \ \wedge \ r \xrightarrow{a} r') \ \Rightarrow \ \exists \tilde{\Sigma} \in \Lambda(\Sigma, a, l(n, m, x)).\ \tilde{\Sigma} \approx r'$*

Proof From $r \xrightarrow{a} r'$ and $\Sigma \approx r$ it follows that there exists a Σ' s.t. $\Sigma \stackrel{\hat{a}}{\Rightarrow} \Sigma'$ and $\Sigma' \approx r'$. Let σ be the sequence leading from Σ to Σ'. By Lemma 14 there is a subsequence $\tilde{\sigma}$ s.t. $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$, $\tilde{\Sigma} \leq_{n(m^2-m)/2} \Sigma'$ and

$$length(\tilde{\sigma}) \leq n^2 * n^{(m^2-m)/2)} + (x + (2^n - 1)n^2 * n^{(m^2-m)/2)})\frac{2^{n^2-n}-1}{2^n-1} =: l(n, m, x)$$

Therefore $\tilde{\Sigma} \in \Lambda(\Sigma, a, l(n, m, x))$.

It remains to prove that $\tilde{\Sigma} \approx r'$. As $\Sigma \approx r$ and $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$ there must be a $\tilde{r} \in R$ s.t. $\tilde{\Sigma} \approx \tilde{r}$. Because of Lemma 7 it now suffices to prove that $r' \approx_{(m^2-m)/2} \tilde{r}$. As $\tilde{r} \approx \tilde{\Sigma} \leq_{n(m^2-m)/2} \Sigma' \approx r'$ the result follows from Lemma 13. □

Definition 16. Let A be a finite set of pairs of the form (r, Σ), where r is a state in a finite-state system R with m states and Σ a marking of a communication-free net of size n. Let $l : \mathbb{N}^3 \to \mathbb{N}$ be the function from Lemma 15. A' is a *bounded weak expansion* of A iff:

- For every pair $(r, \Sigma) \in A$ and every step $r \xrightarrow{a} r'$ there exists a $\Sigma' \in \Lambda(\Sigma, a, l(n, m, x))$ with $(r', \Sigma') \in A'$, where x is the size of Σ.
- For every pair $(r, \Sigma) \in A$ and every step $\Sigma \xrightarrow{a} \Sigma'$ there exists an $r' \in \Lambda(r, a, m - 1)$ with $(r', \Sigma') \in A'$. Note that unlike in the previous case this is no restriction, because R has only m states.
- A' is minimal; no proper subset of A' satisfies these two properties.

Let $bwexp(A)$ be the bounded weak expansions of A. Note that $bwexp(A)$ is finite if A is finite.

[1] This does not necessarily mean that the cycles are done first, and the non-cyclic moves afterwards. Moves belonging to cycles and non-cyclic moves can occur in any order.

Lemma 17. *Let A be a finite set of pairs of the form (r, Σ), where r is a state in a finite-state system, Σ a marking of a communication-free net and $r \approx \Sigma$. Then there is a bounded weak expansion $A' \in bwexp(A)$ s.t. $\forall (r', \Sigma') \in A'. r' \approx \Sigma'$.*

Proof directly from Definition 16 and Lemma 15. □

Theorem 18. *It is decidable if a BPP and a finite-state transition system are weakly bisimilar.*

Proof The general structure of the proof is similar to the proof of the decidability of strong bisimulation equivalence for BPP [2]. Basically the decision algorithm builds a finite tableau. Lemma 6 and Dickson's lemma (5) are used to achieve finite depth and Lemma 17 is used to make sure that the proof tree is finitely branching. Details can be found in [9]. □

4 Model Checking Communication-free Nets

Model checking algorithms fall into two classes: iterative algorithms and tableau-based algorithms. While iterative algorithms compute all the states of the system which have the desired property, tableau-based algorithms are designed to check whether a particular expression has a temporal property. This is called local model checking which avoids the investigation of irrelevant parts of the process being verified. Thus they are applicable to the verification of systems with infinite-state spaces. The algorithm here is tableau-based and decides the truth of a formula for an infinite-state concurrent system by examining only finitely many states.

The branching time temporal logic UB^- of [4] is used to describe properties of Petri-nets N. The syntax of the calculus is as follows:

$$\Phi \stackrel{\text{def}}{=} s \geq k \ \mid \ s \leq k \ \mid \ \neg\Phi \ \mid \ \Phi_1 \wedge \Phi_2 \ \mid \ \Diamond\Phi$$

where s ranges over the places of N and $k \in \mathbb{N}$. Disjunction and another modal operator $\Box$ can be added by defining $\Box := \neg\Diamond\neg$.

Let $\mathcal{F}$ be the set of all formulae. Let Ω be the set of all markings of N. The denotation $||\Phi||$ of a formula Φ is the set of markings of N inductively defined by the following rules:

$$\begin{aligned}
||s \geq k|| &= \{\Sigma \mid \Sigma(s) \geq k\} \\
||s \leq k|| &= \{\Sigma \mid \Sigma(s) \leq k\} \\
||\neg\Phi|| &= \Omega - ||\Phi|| \\
||\Phi_1 \wedge \Phi_2|| &= ||\Phi_1|| \cap ||\Phi_2|| \\
||\Diamond\Phi|| &= \{\Sigma \mid \exists \Sigma \xrightarrow{\sigma} \Sigma'. \Sigma' \in ||\Phi||\}
\end{aligned}$$

$\Sigma \in ||\Phi||$ is also denoted by $\Sigma \models \Phi$. An instance of the model checking problem is a net N with a marking Σ and a formula Φ. The question is if $\Sigma \models \Phi$.

It is possible to add the usual one-step nexttime-operator EX to this logic. For labelled nets it is often denoted as $[a]$, with $a \in Act$ and defined by

$$||[a]\Phi|| = \{\Sigma \mid \exists \Sigma \xrightarrow{a} \Sigma' \in ||\Phi||\}$$

The complexity results of this section carry over to this extended logic.

While model checking with UB^- is undecidable for general Petri nets [4] it is still decidable for communication-free nets [4]. The exact complexity of the problem was unknown so far. We show that this model checking problem for communication-free nets is PSPACE-complete.

Definition 19. $\mathcal{F}_d \subset \mathcal{F}$ is defined as the set of all formulae with a nesting depth of modal operators $\Diamond$ of at most d. (It follows that formulae in $\mathcal{F}_0$ contain no modal operators.)

We show that in order to decide $\Sigma \models \Diamond\Phi$ it suffices to check $\Sigma' \models \Phi$ for those Σ' that can be reached from Σ by a sequence of bounded length.

Lemma 20. *Let N be a communication-free net of size n and Σ_1 and Σ_2 two markings of N. Let $\Phi \in \mathcal{F}_d$ and $\hat{k}$ be the maximal k occurring in a subterm of Φ of the form $s \geq k / s \leq k$. If $\Sigma_1 \leq_{(\hat{k}+1)n^d} \Sigma_2$ then $\Sigma_1 \models \Phi \Leftrightarrow \Sigma_2 \models \Phi$.*

Proof by induction on d.

1. If $d = 0$ then Φ doesn't contain any modal operators and $\Sigma_1 \leq_{(\hat{k}+1)} \Sigma_2$. Thus for all places s and any $k \leq \hat{k}$ $\Sigma_1(s) \geq k \Leftrightarrow \Sigma_2(s) \geq k$ and $\Sigma_1(s) \leq k \Leftrightarrow \Sigma_2(s) \leq k$. By induction on the structure of Φ the result follows.
2. Now $d > 0$. For any subterm ψ of Φ s.t. $\psi \in \mathcal{F}_{d-1}$ the induction hypothesis yields $\Sigma_1 \models \psi \Leftrightarrow \Sigma_2 \models \psi$. It only remains to prove that $\Sigma_1 \models \psi \Leftrightarrow \Sigma_2 \models \psi$ for the minimal subterms ψ of Φ s.t. $\psi \in \mathcal{F}_d - \mathcal{F}_{d-1}$. These subterms are of the form $\psi = \Diamond\varphi$ for a $\varphi \in \mathcal{F}_{d-1}$.
   - $\Rightarrow$ If $\Sigma_1 \models \Diamond\varphi$ then there is a sequence σ s.t. $\Sigma_1 \xrightarrow{\sigma} \Sigma_1'$ and $\Sigma_1' \models \varphi$. By Lemma 12 there is a greater sequence σ' s.t. $\Sigma_2 \xrightarrow{\sigma'} \Sigma_2'$ and $\Sigma_1' \leq_{(\hat{k}+1)n^{(d-1)}} \Sigma_2'$. By induction hypothesis $\Sigma_2' \models \varphi$ and therefore $\Sigma_2 \models \Diamond\varphi$.
   - $\Leftarrow$ If $\Sigma_2 \models \Diamond\varphi$ then there is a sequence σ s.t. $\Sigma_2 \xrightarrow{\sigma} \Sigma_2'$ and $\Sigma_2' \models \varphi$. By Lemma 11 there is a subsequence σ' s.t. $\Sigma_1 \xrightarrow{\sigma'} \Sigma_1'$ and $\Sigma_1' \leq_{(\hat{k}+1)n^{(d-1)}} \Sigma_2'$. By induction hypothesis $\Sigma_1' \models \varphi$ and therefore $\Sigma_1 \models \Diamond\varphi$.

$\square$

Lemma 21. *Let N be a communication-free net of size n, Σ a marking with x tokens, $\Phi \in \mathcal{F}_d$ and $\hat{k}$ be the maximal k in a subterm of Φ of the form $s \geq k / s \leq k$. Then $\Sigma \models \Diamond\Phi \Leftrightarrow \exists \Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}.\ \tilde{\Sigma} \models \Phi \wedge length(\tilde{\sigma}) \leq O((x + \hat{k}) * 2^{n^2} * n^d)$*

Proof There must be a sequence σ s.t. $\Sigma \xrightarrow{\sigma} \Sigma'$ and $\Sigma' \models \Phi$. By Lemma 14 there is a subsequence $\tilde{\sigma}$ s.t. $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$, $\tilde{\Sigma} \leq_{(\hat{k}+1)n^d} \Sigma'$ and

$$length(\tilde{\sigma}) \leq n^2 * ((\hat{k}+1)n^d) + (x + (2^n - 1) * n^2 * (\hat{k}+1)n^d) * \frac{2^{n^2-n} - 1}{2^n - 1}$$

So $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$, $length(\tilde{\sigma}) \leq O((x + \hat{k}) * 2^{n^2} * n^d)$ and by Lemma 20 $\tilde{\Sigma} \models \Phi$. $\square$

Lemma 22. *Let N be a communication-free net, Σ a marking of N and $\Phi \in \mathcal{F}_d$. The problem $\Sigma \models \Diamond\Phi$ can be solved in Σ^p_{d+1}.*

Proof by induction on d. Let n be the size of the whole problem. It follows that N has $\leq n$ places and if $\hat{k}$ is the maximal k occurring in a subterm of Φ of the form $s \leq k/s \geq k$ then $\hat{k} \leq O(2^n)$. Also $x := size(\Sigma) \leq O(2^n)$ and $d \leq O(n)$.

1. If $d = 0$ then Φ doesn't contain any modal operators. By Lemma 21 it suffices to look for a $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$ s.t. $length(\tilde{\sigma}) \leq O((x+\hat{k})2^{n^2} * n^d)$ and $\tilde{\Sigma} \models \Phi$. As $\hat{k} \leq O(2^n)$ and $x \leq O(2^n)$ and $d \leq O(n)$ the Parikh-vector of $\tilde{\sigma}$ can be written in polynomial space. Esparza [5] showed that for communication-free nets it is decidable in polynomial time if there is a fireable sequence of transitions with a given Parikh-vector. (Let P be the Parikh-vector and M the matrix describing the net. There is a fireable sequence σ with vector P iff $\Sigma + M \cdot P \geq \mathbf{0}$ and every nonempty siphon of the subnet generated by the transitions occurring in P is marked by Σ.) Now guess a Parikh-vector and check in polynomial time if there is a $\tilde{\sigma}$ with this Parikh-vector s.t. $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$ and $\tilde{\Sigma} \models \Phi$. It only takes polynomial time to compute the resulting marking $\tilde{\Sigma}$ and to check if $\tilde{\Sigma} \models \Phi$. So the problem can be solved in $NP = \Sigma^p_1$.
2. Now $d > 0$. Again by Lemma 21 it suffices to guess a Parikh-vector of polynomial size. Now check in polynomial time if there is a fireable sequence $\tilde{\sigma}$ with this Parikh-vector s.t. $\Sigma \xrightarrow{\tilde{\sigma}} \tilde{\Sigma}$ and compute $\tilde{\Sigma}$ in polynomial time. As $size(\Sigma) = x$ and $length(\tilde{\sigma}) \leq O((x+\hat{k})2^{n^2} * n^d)$ one can assume that $size(\tilde{\Sigma}) \leq O(x + 2^n((x+\hat{k})2^{n^2} * n^d))$. It follows that $size(\tilde{\Sigma}) \leq O(2^{n^2})$. It is possible to apply the induction hypothesis and check if $\tilde{\Sigma} \models \Phi$ in polynomial time with the help of a Σ^p_d-oracle. Therefore the problem can be solved in $NP^{\Sigma^p_d} = \Sigma^p_{d+1}$.

□

Lemma 23. *Let N be a communication-free net, Σ a marking of N and $\Phi \in \mathcal{F}_d$. The problem $\Sigma \models \Diamond\Phi$ is Σ^p_{d+1}-hard.*

Proof by Esparza in [4]. □

This hardness result even holds for communication-free nets with a finite state space. It remains true if the logic is restricted to statements of the form $s > 0$ instead of $s \geq k/s \leq k$.

Theorem 24. *Let N be a communication-free net, Σ a marking of N and $\Phi \in \mathcal{F}_d$. The problem $\Sigma \models \Diamond\Phi$ is Σ^p_{d+1}-complete.*

Proof directly from Lemma 22 and Lemma 23. □

Theorem 25. *Let N be a communication-free net, Σ a marking of N and $\Phi \in \mathcal{F}$. The problem $\Sigma \models \Phi$ is PSPACE-complete.*

Proof By analyzing the proof of Lemma 22 we show a bound for the size of the subsequently computed markings. $size(\tilde{\Sigma}) \leq O(size(\Sigma) * (2^{n^2} * n^d))$. As only d such markings are computed the maximal size is $O(2^{(n^2+log(n)*d)*d})$. As $d \leq O(n)$ the algorithm is in $NSPACE(O(n^3)) \subseteq PSPACE$. PSPACE-hardness follows from the proof of Lemma 23 as the influence of parameter d is linear. □

5 Conclusion

Basic Parallel Processes are a weak model of concurrent computation. It can be argued that any decent model of concurrent computation should be at least as powerful as BPP. What makes them interesting is that they are a model for infinite state concurrent systems that seems to lie just on the "border of decidability". Some properties that are undecidable for more powerful models of concurrent systems are still decidable for BPP. These include strong bisimulation equivalence [2], weak bisimulation equivalence to a finite-state LTS (see section 3) and model checking with the weak branching time temporal logic UB^- of section 4. On the other hand BPP are powerful enough to make some properties undecidable, like model checking with the modal μ-calculus [4] and language equivalence [6].

Acknowledgement

Many thanks to Javier Esparza for helpful discussions.

References

1. S. Christensen. *Decidability and Decomposition in Process Algebras.* PhD thesis, Edinburgh University, 1993.
2. S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation equivalence is decidable for basic parallel processes. In E. Best, editor, *Proceedings of CONCUR 93*, number 715 in LNCS. Springer Verlag, 1993.
3. S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivelence is decidable for all context-free processes. In W.R. Cleaveland, editor, *Proceedings of CONCUR 92*, number 630 in LNCS. Springer Verlag, 1992.
4. Javier Esparza. Decidability of model checking for infinite-state concurrent systems. *Acta Informatica*, 1995.
5. Javier Esparza. Petri nets, commutative context-free grammars and basic parallel processes. In Horst Reichel, editor, *Fundamentals of Computation Theory*, number 965 in LNCS. Springer Verlag, 1995.
6. Y. Hirshfeld. Petri nets and the equivalence problem. In *Proceedings of CSL'93*, number 832 in LNCS, pages 165–174. Springer Verlag, 1993.
7. P. Jančar. Undecidability of bisimilarity for petri nets and some related problems. *Theoretical Computer Science*, 1995.
8. P. Jančar and F. Moller. Checking regular properties of petri nets. In Insup Lee and Scott A. Smolka, editors, *Proceedings of CONCUR'95*, number 962 in LNCS. Springer Verlag, 1995.
9. Richard Mayr. Some results on basic parallel processes. Technical Report TUM-I9616, TU-München, March 1996.
10. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.