Available online at www.sciencedirect.com
SCIENCE DIRECT
Theoretical Computer Science 360 (2006) 172–192
Theoretical Computer Science
www.elsevier.com/locate/tcs

Decidability of performance equivalence for basic parallel processes☆

Sławomir Lasota*

Institute of Informatics, Warsaw University, 02-097 Warszawa, ul. Banacha 2, Poland

Received 11 November 2004; received in revised form 15 February 2006; accepted 19 February 2006

Communicated by R. Glabbeek

Abstract

We study an extension of the class of Basic Parallel Processes (BPP), in which actions are durational and urgent and parallel components have independent local clocks. The main result is decidability of strong bisimilarity, known also as performance equivalence, in this class. This extends the earlier decidability result for plain BPP by Christensen et al. Our decision procedure is based on decidability of the validity problem for Presburger arithmetic. We prove also polynomial complexity in positive-duration fragment, thus properly extending a previous result by Bérard et al. Both ill-timed and well-timed semantics are treated.

Keywords: Process algebra; Basic parallel processes; Bisimulation equivalence; Performance equivalence; Equivalence checking

1. Introduction

One important problem in verification of concurrent systems is to check whether two given systems P and Q are equivalent under a given notion of equivalence. A lot of research effort has been devoted recently to investigate decidability and complexity of different behavioral equivalences for various process algebras generating infinite-state systems. Since equivalence checking is undecidable in general, some restricted classes of processes have been defined and investigated. We study here Basic Parallel Processes (BPP) [11], an extension of recursively defined finite-state systems by an operator of parallel composition.

Bisimilarity (bisimulation equivalence) plays a central role in the theory of concurrent processes. It is a well accepted behavioral equivalence which often remains decidable for infinite-state systems. An elegant proof of decidability of bisimilarity for BPP was given in [12], while all other equivalences in the well-known van Glabbeek's spectrum are undecidable for this class. Recently, PSPACE-completeness has been established by Srba [29] and Jančar [23].

In order to deal with timing aspects of systems, process algebras were extended with an appropriate notion of time (see e.g. [26,2,20,14,13,3]). Furthermore, a substantial effort have also been directed toward defining a robust notion

☆ This work was partially supported by the Polish KBN grant No. 4 T11C 042 and by the Research Training Network GAMES.
* Tel.: +48 22 5544573; fax: +48 22 5544400.
E-mail address: sl@mimuw.edu.pl.

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.02.020

of equivalence taking performance into consideration, e.g., equivalence that relates only those processes that exhibit the same behavior at the same speed. One successful attempt is bisimilarity introduced in [20] and then investigated in [14,13]. The equivalence was called *performance equivalence* there. (In fact it is strong bisimilarity over a natural time-stamped transition system defined in [5]. We prefer to adopt a different name, following [20,5], in order to avoid confusion with untimed strong bisimilarity.) It applies to discrete-time process algebras where actions are *durational* and *urgent*. Urgent actions take place as soon as possible and can only be delayed in order to wait for a communication with another process. Moreover, it is assumed that parallel components have their own local clocks, i.e., time can pass in different components at different rates. The semantics allows for runs which are *ill-timed*, i.e., the observed time of an action need not to increase along a run, but *well-caused*, i.e., causally related local clock values do increase along a run. Hence, the motivation is not to model real time here, unlike e.g. in [26,3]. The intention is rather to capture effectiveness of parallel computation as precisely as possible.

A timed system is inherently infinite-state in general, even if its untimed analogue is not. Hence the research on equivalence checking of such systems was mostly restricted to systems with finite control, like timed automata [10] or finite-control fragment of timed process algebra [15]. In contrast to this, there are only few results on verification of timed systems with infinite number of control states. We can mention here [28], where the tableau-based technique of [12] was adapted to timed-arc BPP nets.

One relevant result by Berard et al. [5] applies to BPP in a particularly restricted normal form. The language considered there falls within a *positive-duration* fragment, i.e., a subclass with durations of all actions restricted to strictly positive integers. The normal form considered in [5] is although too restrictive for this fragment. On the positive side, surprisingly, the polynomial-time complexity is established. Apparently, the positive-duration fragment does not subsume plain BPP.

As the main result of this paper, we prove decidability of performance equivalence (or strong bisimilarity) for the extension of BPP with arbitrary non-negative durations. The extended language is called *timed BPP* (TBPP) in the sequel. In fact, our formulation of the process algebra is simpler and more general than in previous papers. The duration of an action is not assigned statically as in [20,14,5]. Instead, a duration is associated explicitly with each action prefix appearing in an expression. In particular, duration of an action can be chosen in a non-deterministic way.

Our results extend two previously known decidability results for plain BPP [12] and for the restricted normal form of [5]. In fact, neither of proof methods applied for these two results can be used directly here. On one side, all the known decidability proofs for strong bisimulation equivalence on BPP are based, sometimes indirectly, on the fact that BPP can be seen, up to a structural congruence, as a finitely generated commutative semigroup. As a consequence, bisimilarity on BPP is always finitely presentable and even semi-linear. The only exception is the proof by Jančar [23], but even there semi-linearity of bisimilarity was crucial. All these approaches cannot be translated directly to TBPP, since parallel components of a TBPP process can be, intuitively, arbitrarily spread out over time axis. Our effort in this paper is actually focused on finding again a way of finitely presenting strong bisimulation equivalence (i.e., performance equivalence) on TBPP. This is achieved in Sections 4 and 5.

On the other side, the restriction to positive durations implies a particularly nice decomposition property w.r.t. parallel composition, noticed already in [5], and used as a core of the effective algorithm. In Section 9 we prove the decomposition property for the whole positive-duration fragment. Then we exploit this property to show that performance equivalence coincides with *distributed bisimilarity* [7,11]. Since the latter is decidable in polynomial time [25], we get the same complexity for our problem.

Our result on coincidence of performance equivalence and distributed bisimilarity completes in fact the picture of non-interleaving equivalences on BPP. As mentioned in [17], *location equivalence* [8], *causal equivalence* and distributed bisimilarity all coincide on BPP with communication. Furthermore, causal equivalence and *history preserving bisimilarity* [19] coincide on BPP by the result of Aceto [1]. And finally, Fröschle showed coincidence of distributed and history preserving bisimilarity [17]. Hence all non-interleaving equivalences essentially coincide on BPP, including performance equivalence, with one notable exception—the hereditary history preserving bisimilarity [4].

The decomposition exhibited by the positive-duration fragment does not hold in the general case, where zero durations are allowed. Instead, we are able to show partial decomposition, i.e., decomposition w.r.t. future components only, in Section 4. This consists a fundamental tool in our decision procedure.

In the main part of the paper we work with the ill-timed semantics originating from [20]. This corresponds to local-clocks or local-time perspective on timed processes. But for completeness we also investigate the well-timed variant

of the semantics, that directly models the physical time. Surprisingly, for the well-timed semantics the performance equivalence is much easier (see Section 8). In fact, decidability can be shown in this case even without the technical development of Sections 4 and 5.

Outline: After basic definitions in Section 2, we introduce the *standard form* in Section 3. Both these sections contain an exhaustive discussion on how our setting relates to previous approaches. Then we prove in Section 4 a series of partial decomposition results. They allow us to overcome in Section 5 the main difficulty: we show how performance equivalence can be captured inside a finitely generated commutative semigroup. Then Sections 6 contain the decision procedure, exploiting decidability of the validity problem for Presburger arithmetic [27] in the way suggested by Jančar in [22] for untimed systems. In Sections 7 and 8 we discuss the game characterization of performance equivalence and possible extensions of our decidability result, respectively. We consider communication and *well-timed* semantics. The following Section 9 deals with positive-duration fragment and the last section contains a few remarks.

This is an extended and improved version of the preliminary paper [24]. Only processes in standard form were considered in [24]; the positive-duration fragment and game characterizations were not investigated.

2. Basic parallel processes and performance equivalence

BPP [11] is the extension of finite-state system by an operator of parallel composition. Let *Const* be a finite set of *process constants*, ranged over by X, Y, etc., and let *Act* be a finite set of *actions*, ranges over by a, b, etc. The set of BPP process expressions (processes), ranged over by P, Q, P_1, etc. is given by

$$P \ ::= \ \mathbf{0} \mid X \mid a.P \mid P \| P \mid P + P,$$

where $\mathbf{0}$ denotes the empty process having no action, $a.P$ is an action prefix that performs an action a and continues as P, $\_\|\_$ stands for a parallel composition and $\_ + \_$ denotes a non-deterministic choice. Behavior of each constant X will be defined by some (possibly recursive) equation $X \stackrel{\mathtt{def}}{=} P$.

Before providing semantics, we will need a slight extension on BPP. The extended language, called TBPP in this paper, admits additionally a delay operator $1 \rhd \_$. Process $1 \rhd P$ behaves precisely like P but delayed by one time unit. Hence, the syntax of TBPP is given by extending the BPP syntax with one additional production:

$$P \ ::= \ \ldots \mid 1 \rhd P.$$

In the operational semantics to be presented below, the delay will be used to record the time used by the process up to now. Hence, we will write $t \rhd P$ as a shorthand for $\underbrace{1 \rhd (1 \rhd (\ldots (1 \rhd P) \ldots))}_{t \text{ times}}$, for any non-negative integer $t \in \mathbb{N}$; by $\mathbb{N}$ we denote the set of non-negative integers. For convenience we will also admit $0 \rhd P$ as equivalent to P.

We do not need to distinguish between processes related by a *structural congruence*, i.e., the smallest congruence induced by:

- associativity and commutativity of parallel composition and choice,
- nilpotency and time cancellation for $\mathbf{0}$: $\mathbf{0}\|P = P$, $\mathbf{0} + P = P$, $1 \rhd \mathbf{0} = \mathbf{0}$,
- clock distribution laws [14]:

$$1 \rhd (P_1 \| P_2) = 1 \rhd P_1 \| 1 \rhd P_2, \quad 1 \rhd (P_1 + P_2) = 1 \rhd P_1 + 1 \rhd P_2.$$

These are very natural laws satisfied by most of semantical equivalences (performance equivalence in particular). Hence we feel free, from now on, to understand equality of processes up to the structural congruence.

A *TBPP process definition* consists of a finite set *Const* of process constants, a finite set *Act* of *actions* and a set of defining equations $X \stackrel{\mathtt{def}}{=} P$, one for each constant X. Each process definition Δ induces a labeled transition system, whose states are processes, and whose transitions are labeled by pairs (t, a), with t a natural number and $a \in Act$.

The transitions are derived from the following SOS rules (recall that _‖_ and _ + _ are commutative):

$$\frac{}{a.P \xrightarrow{0,a} P} \qquad \frac{P \xrightarrow{t,a} P'}{1 \rhd P \xrightarrow{t+1,a} 1 \rhd P'}$$

$$\frac{P \xrightarrow{t,a} P'}{P \| Q \xrightarrow{t,a} P' \| Q} \qquad \frac{P \xrightarrow{t,a} P'}{P + Q \xrightarrow{t,a} P'} \tag{1}$$

$$\frac{(X \stackrel{\text{def}}{=} P) \in \Delta \qquad P \xrightarrow{t,a} P'}{X \xrightarrow{t,a} P'}.$$

The transition system obtained is finitely branching despite that it has an infinite set of labels. A transition $P \xrightarrow{t,a} Q$ is to mean "P needed t time units so far and now does a and becomes Q"; or "P does a at time t and becomes Q". According to the rule for $1 \rhd \_$, the delay operator is preserved during the transition and thus can be used to record the value t of the clock associated implicitly to each process. But one should remember that time t is *local*, i.e., not related to the amount of time that elapsed in other parallel components. Following [14,20,5], we allow for *ill-timed runs*, i.e., we *do not* exclude

$$P \xrightarrow{t_1,a_1} P' \xrightarrow{t_2,a_2} P''$$

when $t_1 > t_2$. This is due to the rule for _‖_, which allows for a transition in one parallel component independently from values of clocks in other components. It was argued convincingly in [2] that ill-timed semantics brings no semantical problems since the ill-timed runs are well-caused and on the other hand greatly simplifies the technical treatment. While we agree with the first point, in Section 8 we demonstrate that deciding strong bisimilarity is considerably easier for well-timed semantics.

Performance equivalence [20] is precisely *strong bisimilarity*, as defined below, over the transition system determined by the rules (1). From now on, we use both terms interchangeably. Note that the latter should not be confused with strong bisimilarity of plain (untimed) BPP (cf. the comment at the end of this section).

Given a process definition Δ, two processes P_1 and P_2 are (strongly) bisimilar w.r.t. Δ, denoted $P_1 \sim_\Delta P_2$ (or $P_1 \sim P_2$ when Δ is understood in the context), if they are related by some (strong) bisimulation R, that is a binary relation over processes such that whenever $(P, Q) \in R$, for each $a \in Act$ and $t \geqslant 0$,

- if $P \xrightarrow{t,a} P'$ then $Q \xrightarrow{t,a} Q'$ for some Q' such that $(P', Q') \in R$,
- if $Q \xrightarrow{t,a} Q'$ then $P \xrightarrow{t,a} P'$ for some P' such that $(P', Q') \in R$.

In [5,14], such relations were called *performance relations* and *R-bisimulations*, respectively, and the induced equivalence was called *performance equivalence*. The equivalence resembles *timed bisimilarity*, as defined e.g. in [26,10]. Besides the fundamental difference between local- and global-time semantics (i.e., between ill-timed and well-timed semantics), one additional difference is that instead of observing portions of time that passes between actions, we allow only for observation of a moment in time at which the action is actually performed. This will be inessential in the restricted language identified below in this section, e.g., because non-deterministic choice due to a sole passing of time will not be allowed. Moreover, explicit observation of time passing is reasonable only in global-time approach and would be meaningless here.

Bisimilarity has a game-theoretic characterization. The *Bisimulation Game* is played between two players, Attacker and Defender, on an arena consisting of pairs of processes (P, Q). In each round, Attacker performs a transition from one of processes P, Q, say $P \xrightarrow{t,a} P'$. The obligation of Defender is to answer in the other process with a matching transition, say $Q \xrightarrow{t,a} Q'$, for some Q'. Then the game continues from (P', Q'). If a player is stuck after a finite number of rounds, his opponent wins. Otherwise, i.e. when the play is infinite, Defender wins. The classical result says that $P \sim Q$ iff Defender has a winning strategy in the game starting in (P, Q).

By now we have allowed the delay operator to appear arbitrarily in process expressions. But in fact the underlying intuition is that the only durational activity of the process is performing some action. Hence, following [5,14,20] we impose a restriction on process expressions. From now on we will assume that in any right-hand side expressions in Δ, delay can only appear immediately after action prefix. Formally, we restrict syntax of the right-hand side expressions

as follows:

$$P \ ::= \ \mathbf{0} \mid X \mid a.t \rhd P \mid P \| P \mid P + P. \tag{2}$$

Hence, each delay $t \rhd \_$ appearing in the expression describes a duration of some action a. We allow for any non-negative $t \in \mathbb{N}$, including $t = 0$, i.e., actions of duration zero. A process expression conforming to the syntax (2) will be called *pure*. Moreover, we will call Δ a *BPP process definition* if only zero-delays $0 \rhd \_$ appear in the right-hand side expressions. The delays t can be encoded either in binary or in unary—this will have no impact on our decidability and complexity results.

Furthermore, following the standard lines, we assume that each right-hand side process is *guarded*, i.e., a constant may only appear in the scope of some action prefix. Such Δ will be called *guarded* as well.

Our restricted syntax (2) is motivated by previous papers, but it is slightly more general. In [20,14,5] the process definition was additionally equipped with an arbitrary *duration function* $f : Act \to \mathbb{N} \setminus \{0\}$, assigning a positive duration to each action. Consequently, instead of our first rule in (1), a rule

$$\frac{}{a.P \xrightarrow{0,a} f(a) \rhd P}$$

was used explicitly in [14,20] and implicitly in [5]. Our approach is more general in at least two respects: first, zero-durations are allowed; second, a duration is assigned to each action prefix separately rather then fixed for all appearances of each action. Furthermore, our setting admits *standard form*, which was not the case in the cited papers. This is so because we may separate in the language an action prefix from specification of its duration. We will further comment on this in Section 3.

In Sections 4–8 we investigate decidability of the following problem:

Problem: STRONG BISIMILARITY FOR TBPP
Instance: A pure guarded TBPP process definition Δ and constants X, Y
Question: $X \sim_\Delta Y$?

We do not lose generality by only restricting to constants, since checking $P \sim Q$, for arbitrary pure and guarded P, Q is equivalent to checking $X_P \sim X_Q$, for two new fresh constants with defining equations: $X_P \stackrel{\text{def}}{=} P$ and $X_Q \stackrel{\text{def}}{=} Q$.

Note that in a special case when Δ is a BPP processes definition, each transition derived from rules (1) is performed at time $t = 0$. Hence the time becomes irrelevant and we essentially obtain a strong bisimilarity problem for plain (untimed) BPP. This means that the lower bound by Srba [29] applies and hence performance equivalence is PSPACE-hard.

In Section 9 we will consider a positive-duration fragment, where a restriction $t > 0$ is imposed on the syntax (2). An expression (2) satisfying this restriction will be called *positive-duration expression* later on; accordingly, Δ will be also called *positive-duration process definition*. This fragment is interesting since it corresponds closely to the semantics of [20,14]. Moreover it admits a polynomial-time algorithm, in contrasts to the general case.

3. Standard form

It is well known [11] that in untimed case, each guarded BPP process definition can be transformed to a *standard form* consisting of equations of the form

$$X \stackrel{\text{def}}{=} a_1.P_1 + \cdots + a_k.P_k, \tag{3}$$

where each of $P_1, \ldots, P_k$ is a parallel composition of constants (possibly empty). Assuming that Δ is pure and guarded, we provide a similar transformation for TBPP; the only difference is that each of $P_1, \ldots, P_k$ is a parallel composition of delayed constants $t \rhd X$ (including possibly $t = 0$).

Applying our structural congruence laws we can group constants with the same delay, which leads to the following *timed normal form*:

$$t_1 \rhd \alpha_1 \| t_2 \rhd \alpha_2 \| \ \cdots \ \| t_n \rhd \alpha_n, \tag{4}$$

where $n \geqslant 0$, $0 \leqslant t_1 < t_2 < \cdots < t_n$ and each of $\alpha_1 \ldots \alpha_n$ is a non-empty parallel composition of constants. When $n = 0$, (4) denotes the empty process $\mathbf{0}$. We assume here and later that delay operator binds stronger than parallel composition.

Theorem 1. *Any pure and guarded process definition Δ can be effectively transformed to a process definition $\bar{\Delta}$ with each right-hand side in standard form* (3) *such that each of $P_1, \ldots, P_k$ are in timed normal form* (4). *Moreover, $\bar{\Delta}$ is equivalent to Δ: for each constant X of Δ there is a constant Z_X in $\bar{\Delta}$ such that $X \sim_{\Delta \cup \bar{\Delta}} Z_X$.*

Proof (*Sketch*). For the sake of transformation, we introduce a fresh constant Z_P for each *pure* subexpression P of each right-hand side, with the defining equation $Z_P \stackrel{\text{def}}{=} P$. Restricting here to only pure subexpressions means that we do not allow to split the prefix-delay pairs $a.t \rhd \_$. Apparently, the newly introduced defining equations may be unguarded. Hence each such equation $Z_P \stackrel{\text{def}}{=} P$ is then transformed to the standard form, as described below.

Consider any transition $P \xrightarrow{0,a} P'$, derived by the SOS rules (1) from a transition $a.t \rhd P'' \xrightarrow{0,a} t \rhd P''$ of a subexpression $a.t \rhd P''$. The latter is a subexpression of P or a subexpression of the definition of an unguarded constant in P. The resulting expression P' is always of the form

$$P' = (t) \rhd P'' \| Q_1 \| \cdots \| Q_l.$$

Moreover, by inspecting the rules (1) we conclude that each Q_i is always a subexpression of some right-hand side, as well as P''. Furthermore, all Q_i and P'' are pure. Hence, we have $P' \sim P'''$, where

$$P''' = (t) \rhd Z_{P''} \| Z_{Q_1} \| \ldots \| Z_{Q_l}. \tag{5}$$

Now, the new definition of the constant Z_P is

$$Z_P \stackrel{\text{def}}{=} a_1.P_1''' + \cdots + a_k.P_k''',$$

where $a_1, \ldots, a_k$ are all possible transitions of P and $P_1''', \ldots, P_k'''$ are derived as in (5) above.

Note that our transformation preserves $\sim$: we only use syntactical congruence laws and substitution of a definition in place of a constant. The resulting process definition is of polynomial size, since only a polynomial number of new constants is introduced, and each new definition is also polynomial in size. The latter is guaranteed since Δ was assumed to be guarded. □

Remark. A process (5) is in a simpler format than (4), but in the following sections we consider the normal form (4), for two reasons. First, (5) changes when communication is allowed (cf. Section 8). Second, and more important, we will need a normal form preserved by transitions: a transition performed by a process in timed normal form always yields a process in timed normal form, when Δ is in standard form.

Note that the process definition in standard form (3) is not pure in general.

Observe that if process definition is in standard form (3), the behavior of a constant X could be equivalently given by a set of rewrite rules:

$$X \xrightarrow{0,a_i} P_i, \quad \text{for } i = 1 \ldots k, \tag{6}$$

instead of the rules for prefix and choice in (1). For convenience, we will assume that Δ contains indeed such rules; and will write $X \xrightarrow{a_i} P_i \in \Delta$, omitting the time-stamp always equal to 0.

In [5] a standard form similar to (3) was assumed, together with rules similar to (6). But since a positive duration was assigned to each action a, the rewrite rules of [5] had, compared to (6), a more restrictive form:

$$X \xrightarrow{0,a_i} f(a_i) \rhd P_i \quad \text{for } i = 1 \ldots k. \tag{7}$$

A serious consequence of this choice is that even the positive-duration fragment is not representable in general by the rewrite rules (7), since (7) always imposes delay of *at least* $f(a_i)$ time units on all process constants in P_i. Consider for instance a very simple counter-example:

$$X \stackrel{\text{def}}{=} a.1 \rhd X \| b.2 \rhd X.$$

In other words, the *expansion law* does not hold, if durations are specified implicitly by a duration function. This is one of the reasons why we do not assume a duration function and prefer to specify duration of an action explicitly in expressions. As a consequence, our polynomial-delay complexity for positive-duration fragment, proved in Section 9, properly extends the result of [5].

A plain BPP process in normal form is a parallel composition of constants. Due to associativity, commutativity and nilpotency laws, it can be seen as a finite multiset of process constants, or equivalently as an element of the free commutative semigroup over the set *Const* of constants. This observation was a key point in the decision procedure for strong bisimilarity of BPP [12]. Unfortunately, in TBPP we lose this finite-generation property: TBPP processes can be rather seen as elements of the free commutative semigroup over $\mathbb{N} \times \mathit{Const}$. This is illustrated by:

Example 1. Assume that two processes $X \| 3 \rhd (Y \| Z)$ and $(Y \| Y) \| 3 \rhd (Z \| Z \| Z)$ are bisimilar. Decomposition results stated in Lemmas 1 and 2 in Section 4 imply that $3 \rhd (Y \| Z)$ and $3 \rhd (Z \| Z \| Z)$ are then bisimilar too. So we can substitute one for another and consider $X \| 3 \rhd (Y \| Z)$ and $(Y \| Y) \| 3 \rhd (Y \| Z)$, which are *almost identical* according to terminology from the next section, i.e., differ only at time 0. At first sight, this short deduction suggests that we are not very far from the bisimilarity checking problem for BPP, as we only need to consider BPP processes X and $Y \| Y$ in a context $\_ \| 3 \rhd (Y \| Z)$, roughly. But this is not really true. Assume that Z has a rewrite rule $Z \xrightarrow{a} Y \| Z$. According to Lemma 5 in the next section, a transition in the context part can always be matched by the identical transition in the context part, e.g., $X \| 3 \rhd (Y \| Z) \xrightarrow{3,a} X \| 3 \rhd (Y \| Y \| Z)$ is matched by $(Y \| Y) \| 3 \rhd (Y \| Z) \xrightarrow{3,a} (Y \| Y) \| 3 \rhd (Y \| Y \| Z)$. But nevertheless we must take into account an infinite number of contexts: $\_ \| 3 \rhd (Y^i \| Z)$, for $i > 0$. Moreover, if Y has for instance a rule $Y \xrightarrow{b} Y \| 1 \rhd Y$, contexts that are arbitrarily spread out through the time domain $\mathbb{N}$ must be considered.

Hence, the proof methods used for decidability of bisimilarity for BPP work no more for TBPP. This difficulty is overcome in Section 5, where we prove that one can restrict to a sufficiently large initial fragment of the time domain and essentially ignore transitions which lead out of this fragment.

4. Decomposition and almost-identical pairs

From now on we assume that Δ is an arbitrary fixed set of rewrite rules of the form (6). Hence, we consider processes in timed normal form (4) only. For such $P = t_1 \rhd \alpha_1 \| t_2 \rhd \alpha_2 \| \ \ldots \ \| t_n \rhd \alpha_n$, which is *non-empty*, i.e., $n > 0$, we put:

- $\mathrm{head}(P) := t_1 \rhd \alpha_1$,
- $\mathrm{tail}(P) := t_2 \rhd \alpha_2 \| \ \ldots \ \| t_n \rhd \alpha_n$,
- $\mathrm{minclock}(P) := t_1$,
- $\mathrm{now}(P) := \alpha_1 \| (t_2 - t_1) \rhd \alpha_2 \| \ \ldots \ \| (t_n - t_1) \rhd \alpha_n$.

Note that $\mathrm{head}(P)$, $\mathrm{tail}(P)$ and $\mathrm{now}(P)$ are in timed normal form again, $\mathrm{head}(\mathrm{now}(P))$ is untimed, $P = \mathrm{head}(P) \| \mathrm{tail}(P)$ and $P = \mathrm{minclock}(P) \rhd \mathrm{now}(P)$.

Bisimilarity is not only compositional (i.e. a congruence) w.r.t. parallel composition and the delay operator but it is also partially decompositional:

Lemma 1 (*Decomposition w.r.t.* $1 \rhd \_$). *For non-empty P and Q,* $P \sim Q$ *implies*
(1) $\mathrm{minclock}(P) = \mathrm{minclock}(Q)$ *and*
(2) $\mathrm{now}(P) \sim \mathrm{now}(Q)$.

Proof. (1) $\mathrm{minclock}(P) = \mathrm{minclock}(Q)$ as the earliest action of P must be matched in Q and vice versa. (2) The relation $\{(t \rhd \mathrm{now}(P), t \rhd \mathrm{now}(Q)) : t \geqslant 0,\ P \sim Q\}$ is a bisimulation. □

Lemma 2 (*Partial decomposition w.r.t.* $\_ \| \_$). *For non-empty P and Q,* $P \sim Q$ *implies* $\mathrm{tail}(P) \sim \mathrm{tail}(Q)$.

Proof. The relation $\{(\mathrm{tail}(P), \mathrm{tail}(Q)) : P \sim Q,\ P \text{ and } Q \text{ are non-empty}\}$ is a bisimulation, since any action in $\mathrm{tail}(P)$ at time $t > \mathrm{minclock}(P) = \mathrm{minclock}(Q)$ must be matched in $\mathrm{tail}(Q)$. □

Example 2. Strong bisimilarity is not fully decompositional w.r.t. $\_\|\_$, $P \sim Q$ does not imply $\mathrm{head}(P) \sim \mathrm{head}(Q)$. As a counterexample, consider a process definition:

$$A \xrightarrow{a} 1 \rhd B, \quad A' \xrightarrow{a} 1 \rhd C, \quad B \xrightarrow{b} B, \quad C \xrightarrow{c} C.$$

We have $A\|1 \rhd (B\|C) \sim A'\|1 \rhd (B\|C)$ but $A \not\sim A'$.

By Lemmas 1 and 2, if two processes are bisimilar, say $t_1 \rhd \alpha_1\| \ \dots \ \|t_n \rhd \alpha_n$ and $u_1 \rhd \beta_1\| \ \dots \ \|u_m \rhd \beta_m$, then necessarily they involve the same time stages, that is to say $n = m$ and $t_i = u_i$, for $i \leqslant n$.

Assumption. In Sections 4, 5 and 6, whenever we consider a pair (P, Q) of processes, we silently assume that $P = t_1 \rhd \alpha_1\| \ \cdots \ \|t_n \rhd \alpha_n$ and $Q = t_1 \rhd \beta_1\| \ \cdots \ \|t_n \rhd \beta_n$, for some $n \geqslant 0$, $0 \leqslant t_1 < t_2 < \cdots < t_n$ and non-empty untimed processes $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$. We omit routine verification that all pairs mentioned throughout the following sections are of this form.

Lemma 2 suggests to study pairs of processes in a particularly simple form:

Definition 1 (*Almost-identical pairs*). Processes P and Q are called *almost-identical* if they are both empty, or $\mathrm{minclock}(P) = \mathrm{minclock}(Q) = 0$ and $\mathrm{tail}(P) = \mathrm{tail}(Q)$.

We will apply Lemma 2 to decompose a pair of processes into a number of almost-identical pairs. As a consequence, we will be able to ignore all pairs which are not almost-identical (Lemma 3). Before the formal definition of decomposition, we give an example to explain the idea.

Example 3. Consider a pair:

$$\begin{aligned} P &= 2 \rhd (X\|Y) \ \| \ 7 \rhd (X\|X\|Y) \ \| \ 10 \rhd (Z\|Z) \ \| \ 16 \rhd Y, \\ Q &= 2 \rhd (X\|X\|Z) \ \| \ 7 \rhd (X\|Z) \ \| \ 10 \rhd (X\|X) \ \| \ 16 \rhd Y. \end{aligned}$$

(P, Q) can be decomposed into four pairs (P_i, Q_i), $i \leqslant 4$. The rule is as follows: $\mathrm{head}(P_i)$ and $\mathrm{head}(Q_i)$ is the ith component of P and Q, respectively, while $\mathrm{tail}(P_i) = \mathrm{tail}(Q_i)$ contains the components of P from all later time stages.

$$\begin{aligned} P_1 &= (X\|Y) \ \| \ 5 \rhd (X\|X\|Y) \ \| \ 8 \rhd (Z\|Z) \ \| \ 14 \rhd Y, \\ Q_1 &= (X\|X\|Z) \ \| \ 5 \rhd (X\|X\|Y) \ \| \ 8 \rhd (Z\|Z) \ \| \ 14 \rhd Y, \\ P_2 &= (X\|X\|Y) \ \| \ 3 \rhd (Z\|Z) \ \| \ 9 \rhd Y, \\ Q_2 &= (X\|Z) \ \| \ 3 \rhd (Z\|Z) \ \| \ 9 \rhd Y, \\ P_3 &= (Z\|Z) \ \| \ 6 \rhd Y, \\ Q_3 &= (X\|X) \ \| \ 6 \rhd Y, \\ P_4 &= Y, \\ Q_4 &= Y. \end{aligned}$$

By Lemmas 1 and 2, $P \sim Q \iff \forall i \leqslant 4. P_i \sim Q_i$. The last pair is identical and hence can be ignored, therefore to show $P \sim Q$ it is sufficient to consider only the first three: $P \sim Q \iff \forall i \leqslant 3. P_i \sim Q_i$. The formal definition follows.

Definition 2 (*Decomposition set*). A *decomposition set* $\mathbb{DS}(P, Q)$ of a pair (P, Q) is defined recursively by (for brevity, $\bar{P}$ stands for $\mathrm{now}(P)$ and $\bar{Q}$ stands for $\mathrm{now}(Q)$)

$$\mathbb{DS}(P, Q) := \begin{cases} \{(\bar{P}, \mathrm{head}(\bar{Q})\|\mathrm{tail}(\bar{P}))\} \cup \mathbb{DS}(\mathrm{tail}(\bar{P}), \mathrm{tail}(\bar{Q})) & \text{if } P \neq Q, \\ \emptyset & \text{if } P = Q. \end{cases}$$

Note that all pairs in $\mathbb{DS}(P, Q)$ are almost-identical by definition.

Lemma 3. *$P \sim Q$ iff $\mathbb{DS}(P, Q) \subseteq \sim$.*

Proof. By induction on the size of P, Q. As for $P = Q$ we are immediately done, assume that P, Q are non-identical, hence non-empty and that the lemma holds for a pair $(\mathrm{tail}(\mathrm{now}(P)), \mathrm{tail}(\mathrm{now}(Q)))$, i.e.,

$$\mathrm{tail}(\mathrm{now}(P)) \sim \mathrm{tail}(\mathrm{now}(Q)) \iff \mathbb{DS}(\mathrm{tail}(\mathrm{now}(P)), \mathrm{tail}(\mathrm{now}(Q))) \subseteq \sim. \tag{8}$$

IF-IMPLICATION: If $\mathbb{DS}(P, Q) \subseteq \sim$, then also $\mathbb{DS}(\mathrm{tail}(\mathrm{now}(P)), \mathrm{tail}(\mathrm{now}(Q))) \subseteq \sim$, as a subset of $\mathbb{DS}(P, Q)$. Hence, by induction assumption (8),

$$\mathrm{tail}(\mathrm{now}(P)) \sim \mathrm{tail}(\mathrm{now}(Q)). \tag{9}$$

Moreover, from $\mathbb{DS}(P, Q) \subseteq \sim$ we derive

$$\mathrm{now}(P) \sim \mathrm{head}(\mathrm{now}(Q)) \| \mathrm{tail}(\mathrm{now}(P)), \tag{10}$$

since this pair is in $\mathbb{DS}(P, Q)$. By (9) and (10) and the congruence property of $\sim$ we conclude $\mathrm{now}(P) \sim \mathrm{now}(Q)$ hence $P \sim Q$.

ONLY-IF-IMPLICATION: Assume $P \sim Q$. By Lemmas 1 and 2, (9) holds again. By (8), $\mathbb{DS}(\mathrm{tail}(\mathrm{now}(P)), \mathrm{tail}(\mathrm{now}(Q))) \subseteq \sim$. Now, we only need to show (10), which follows by (9) and by $\sim$ being a congruence:

$$\mathrm{now}(P) \sim \mathrm{now}(Q) = \mathrm{head}(\mathrm{now}(Q)) \| \mathrm{tail}(\mathrm{now}(Q)) \sim \mathrm{head}(\mathrm{now}(Q)) \| \mathrm{tail}(\mathrm{now}(P)). \qquad \square$$

In Definitions 3, 5 and 6 we introduce a series of computationally more and more tractable notions of bisimulation.

Definition 3 (*Decompositional bisimulation*). A binary relation R over processes is a *decompositional bisimulation* if whenever $(P, Q) \in R$, for each $a \in Act$ and $t \geqslant 0$,

- if $P \xrightarrow{t,a} P'$ then $Q \xrightarrow{t,a} Q'$ for some Q' such that $\mathbb{DS}(P', Q') \subseteq R$,
- if $Q \xrightarrow{t,a} Q'$ then $P \xrightarrow{t,a} P'$ for some P' such that $\mathbb{DS}(P', Q') \subseteq R$.

Lemma 4. *$P \sim Q$ iff P and Q are related by a decompositional bisimulation.*

Before the proof, we need a definition of a *bisimulation-base*. It is an adaptation of the well-known notion due to Caucal [9,11,6].

Definition 4. For a binary relation R, let $\overset{R}{\equiv}$ denote the smallest congruence w.r.t. parallel composition and delay operator containing R. We call R a *bisimulation-base* if whenever $(P, Q) \in R$, for each $a \in Act$ and $t \geqslant 0$,

- if $P \xrightarrow{t,a} P'$ then $Q \xrightarrow{t,a} Q'$ for some Q' such that $(P', Q') \in \overset{R}{\equiv}$,
- if $Q \xrightarrow{t,a} Q'$ then $P \xrightarrow{t,a} P'$ for some P' such that $(P', Q') \in \overset{R}{\equiv}$.

Proof of Lemma 4. By Lemma 3, $\sim$ is a decompositional bisimulation. Hence we only need to show the IF-direction. This is shown in two steps:

(1) each decompositional bisimulation is a bisimulation-base,

(2) if R is a bisimulation-base, then $\overset{R}{\equiv}$ is a bisimulation.

Point (1) is immediate, since each pair (P', Q') belongs to the congruence generated by its decomposition set, $(P', Q') \in \overset{\mathbb{DS}(P', Q')}{\equiv}$. The proof of (2) amounts to showing that the two clauses in the definition of bisimulation hold for $\overset{R}{\equiv}$. This can be done by a straightforward induction on the depth of inference of $P \overset{R}{\equiv} Q$ (see [9]). $\square$

Bisimilarity is closed under identical transitions performed in identical parts of almost-identical processes:

Lemma 5. *Whenever P and Q are almost-identical, $P \sim Q$ and $\mathrm{tail}(P) \xrightarrow{t,a} P'$ then $\mathrm{head}(P) \| P' \sim \mathrm{head}(Q) \| P'$.*

Proof. Note that, by assumption, t is necessarily strictly positive. Thus the answer of Q to the transition $P \xrightarrow{t,a} \mathrm{head}(P) \| P'$ is performed in $\mathrm{tail}(Q)$, since the time t must be matched exactly. Hence for some Q', $\mathrm{tail}(Q) \xrightarrow{t,a} Q'$ and $\mathrm{head}(P) \| P' \sim \mathrm{head}(Q) \| Q'$. By Lemma 2 and the congruence property of $\sim$, $\mathrm{head}(Q) \| Q' \sim \mathrm{head}(Q) \| P'$, which completes the proof. □

To make our notation more succinct, we introduce a closure operator on binary relations over processes as follows: let $R^{\rightarrow}$ be the smallest relation containing R such that whenever an almost-identical pair (P, Q) is in $R^{\rightarrow}$ and $\mathrm{tail}(P) \xrightarrow{t,a} P'$ (or $\mathrm{tail}(Q) \xrightarrow{t,a} P'$) then $(\mathrm{head}(P) \| P', \mathrm{head}(Q) \| P') \in R^{\rightarrow}$. Lemma 5 can be then shortly rephrased by $\sim^{\rightarrow} \subseteq \sim$.

Due to Lemma 3, it is sufficient to consider almost-identical pairs. Lemma 5 allows us to refine the notion of decompositional bisimulation for almost-identical pairs in Definition 5 and to strengthen Lemma 4 in Lemma 6.

Definition 5. A binary relation R over processes is an *almost-identical decompositional bisimulation* if it only contains almost-identical pairs and

(1) whenever $(P, Q) \in R$, for each $a \in Act$,
- if $P \xrightarrow{0,a} P'$ then $Q \xrightarrow{0,a} Q'$ for some Q' such that $\mathbb{DS}(P', Q') \subseteq R$,
- if $Q \xrightarrow{0,a} Q'$ then $P \xrightarrow{0,a} P'$ for some P' such that $\mathbb{DS}(P', Q') \subseteq R$,

(2) $R^{\rightarrow} \subseteq R$.

Roughly, the Attacker's invention in the Bisimulation Game is restricted only to moves at time 0, since all transitions performed at time $t > 0$ are matched identically.

Lemma 6. *Let P and Q be almost-identical. $P \sim Q$ iff P and Q are related by an almost-identical decompositional bisimulation.*

Proof. By Lemma 5, $\sim$ restricted to only almost-identical pairs is an almost-identical decompositional bisimulation. On the other hand, each almost-identical decompositional bisimulation is obviously a decompositional bisimulation. □

5. Bounded bisimulations

Having elaborated an appropriate notion of bisimulation, we are ready to face the main difficulty of the problem. It turns out that we can restrict ourselves to only a finite initial fragment of the time domain $\mathbb{N}$, without losing the completeness of almost-identical decompositional bisimulations w.r.t. $\sim$ as stated in Lemma 6. The restriction is imposed by a "horizon" $2{\cdot}H$, where H is defined as

$$H := \max\{\mathrm{maxclock}(P) : X \xrightarrow{0,a} P \in \Delta\}$$

and $\mathrm{maxclock}(P)$ of a non-empty process P in timed normal form (4) is to denote t_n and $\mathrm{maxclock}(\mathbf{0}) = 0$. Intuitively, H is the maximal *scope* of a single transition: in effect of a transition performed at time t, say, only process components at time between t and $t+H$ can be modified. Recall that according to our transformation to standard form in Section 3, H equals the greatest duration of an action.

Definition 6. A binary relation R over processes is a *bounded almost-identical decompositional bisimulation* (*bounded bisimulation* in short) if it only contains almost-identical pairs (P, Q) with $\mathrm{maxclock}(P) \leqslant 2{\cdot}H$ and $\mathrm{maxclock}(Q) \leqslant 2{\cdot}H$, and whenever $(P, Q) \in R$, for each $a \in Act$,

(1)
- if $P \xrightarrow{0,a} P'$ then $Q \xrightarrow{0,a} Q'$ for some Q' such that $\mathbb{DS}(P', Q') \subseteq R$
- if $Q \xrightarrow{0,a} Q'$ then $P \xrightarrow{0,a} P'$ for some P' such that $\mathbb{DS}(P', Q') \subseteq R$

(2) for $0 < t \leqslant H$, if $\mathrm{tail}(P) \xrightarrow{t,a} P'$ (or $\mathrm{tail}(Q) \xrightarrow{t,a} P'$) then $(\mathrm{head}(P) \| P', \mathrm{head}(Q) \| P') \in R$.

Having restricted in Lemma 6 non-identical responses of the Defender to transitions at $t{=}0$, we restrict ourselves further and take into consideration only transitions at time $t \leqslant H$. Intuitively, H is chosen sufficiently large to guarantee that the additional restriction has no impact on the relevant part of the Bisimulation Game played at time 0. If one reminds that the scope of transitions performed at time 0 does not go beyond $0 \ldots H$, then all transitions (and the identically matching responses) that are performed at time $t > 0$ belonging to that scope are clearly taken into consideration. This guarantees that each bounded bisimulation can be extended to an almost-identical decompositional bisimulation, by closure under identically matched transitions performed at time $t > H$.

Lemma 7. *Let P and Q be almost-identical with* $\mathrm{maxclock}(P) \leqslant 2{\cdot}H$. *Then $P \sim Q$ iff P and Q are related by a bounded bisimulation.*

Proof. ONLY-IF-IMPLICATION: A subset of $\sim$, namely $\sim$ restricted to only those almost-identical pairs (P, Q) with $\mathrm{maxclock}(P) \leqslant 2{\cdot}H$, is a bounded bisimulation.

IF-IMPLICATION: Relying on Lemma 6, we will show that if R is a bounded bisimulation then $R^{\rightarrow}$ is an almost-identical decompositional bisimulation. Point (2) in Definition 5 is immediate, i.e., $(R^{\rightarrow})^{\rightarrow} \subseteq R^{\rightarrow}$. For the proof of Point (1). we will need the following notation. For $P = t_1 \rhd \alpha_1 \| \ \ldots \ \| t_n \rhd \alpha_n$ and $t \geqslant 0$, let

$$\mathrm{head}_t(P) = t_1 \rhd \alpha_1 \| \ \cdots \ \| t_i \rhd \alpha_i,$$
$$\mathrm{tail}_t(P) = t_{i+1} \rhd \alpha_{i+1} \| \ \cdots \ \| t_n \rhd \alpha_n,$$

where i is chosen so that $t_j <= t$ for all $1 \leqslant j \leqslant i$ and $t_j > t$ for all $i+1 \leqslant j \leqslant n$. In particular $\mathrm{head}(P) = \mathrm{head}_0(P)$ and $\mathrm{tail}(P) = \mathrm{tail}_0(P)$, when $\mathrm{minclock}(P) = 0$.

We need to show Point (1) in Definition 5 only for pairs in $R^{\rightarrow} \setminus R$. Let (P, Q) be any pair in $R^{\rightarrow} \setminus R$. So there exists a pair $(P_0, Q_0) \in R$ such that P is obtained from P_0 and Q is obtained from Q_0 by a finite sequence of transitions performed in $\mathrm{tail}(P_0) = \mathrm{tail}(Q_0)$. W.l.o.g. we can assume that the transitions performed at time less or equal to H precede in the sequence those performed at time strictly greater than H. Thus we can further assume that all transitions in the sequence are performed at time greater than H, since all earlier transitions performed at time less or equal H do not lead out of R. To be more precise, there is $(P_0, Q_0) \in R$ such that

$$\mathrm{head}_H(P_0) = \mathrm{head}_H(P), \quad \mathrm{head}_H(Q_0) = \mathrm{head}_H(Q), \tag{11}$$

and there exists $m > 0$, $t_1, \ldots, t_m > H$ and $a_1, \ldots, a_m \in Act$ such that

$$\mathrm{tail}_H(P_0) \xrightarrow{t_1,a_1} \cdots \xrightarrow{t_m,a_m} \mathrm{tail}_H(P). \tag{12}$$

Recall that both (P_0, Q_0) and (P, Q) are almost-identical pairs, hence

$$Q_0 = \mathrm{head}_H(Q_0) \| \mathrm{tail}_H(P_0) \quad \text{and} \quad Q = \mathrm{head}_H(Q) \| \mathrm{tail}_H(P).$$

Now we will show only the first clause in Point (1)—the second clause follows by the identical reasoning. Here is the simple idea underlying the pedantic analysis to follow: all transitions performed at time 0 by P or Q have its scope inside $\mathrm{head}_H(P)$ or $\mathrm{head}_H(Q)$, respectively, hence by (11) the matching of a transition of P_0 by Q_0 can be re-used for P and Q. Assume that P has a transition labeled by $(0, a)$. This means that $\mathrm{head}_H(P) \xrightarrow{0,a} P'$ for some P' with $\mathrm{maxclock}(P') \leqslant H$ and $P \xrightarrow{0,a} P' \| \mathrm{tail}_H(P)$. Hence $\mathrm{head}_H(P) \| \mathrm{tail}_H(P_0) = P_0 \xrightarrow{0,a} P' \| \mathrm{tail}_H(P_0)$.

$$\begin{array}{ccccc}
\overbrace{\mathrm{head}_H(P) \| \mathrm{tail}_H(P_0)}^{P_0} & \xrightarrow{t_1,a_1} \cdots \xrightarrow{t_m,a_m} & \overbrace{\mathrm{head}_H(P) \| \mathrm{tail}_H(P)}^{P} \\
\downarrow{\scriptstyle 0,a} & & \downarrow{\scriptstyle 0,a} \\
P' \| \mathrm{tail}_H(P_0) & \xrightarrow{t_1,a_1} \cdots \xrightarrow{t_m,a_m} & P' \| \mathrm{tail}_H(P)
\end{array}$$

$(P_0, Q_0) \in R$ and R is a bounded bisimulation, hence there exists a Q' such that $\mathrm{maxclock}(Q') \leqslant H$, $\mathrm{head}_H(Q_0) \xrightarrow{0,a} Q'$ and

$$\mathbb{DS}(P' \| \mathrm{tail}_H(P_0), Q' \| \mathrm{tail}_H(P_0)) \subseteq R. \tag{13}$$

This means that $Q \xrightarrow{0,a} Q'\|\mathrm{tail}_H(P)$ and we have the following picture again:

$$\begin{array}{ccccc} Q'\|\mathrm{tail}_H(P_0) & & \xrightarrow{t_1,a_1} \cdots \xrightarrow{t_m,a_m} & & Q'\|\mathrm{tail}_H(P) \\ \uparrow{\scriptstyle 0,a} & & & & \uparrow{\scriptstyle 0,a} \\ \underbrace{\mathrm{head}_H(Q)\|\mathrm{tail}_H(P_0)}_{Q_0} & & \xrightarrow{t_1,a_1} \cdots \xrightarrow{t_m,a_m} & & \underbrace{\mathrm{head}_H(Q)\|\mathrm{tail}_H(P)}_{Q} \end{array}$$

We only need to show that

$$\mathbb{DS}(P'\|\mathrm{tail}_H(P),\, Q'\|\mathrm{tail}_H(P)) \subseteq R^{\rightarrow}.$$

Consider any pair $(\bar{P},\, \bar{Q}) \in \mathbb{DS}(P'\|\mathrm{tail}_H(P),\, Q'\|\mathrm{tail}_H(P))$. By (13) it is sufficient to show that there exists a pair $(\bar{P}_0,\, \bar{Q}_0) \in \mathbb{DS}(P'\|\mathrm{tail}_H(P_0),\, Q'\|\mathrm{tail}_H(P_0))$ such that

$$\mathrm{tail}(\bar{P}_0) = \mathrm{tail}(\bar{Q}_0) \xrightarrow{\bar{t_1},\bar{a_1}} \cdots \xrightarrow{\bar{t_m},\bar{a_m}} \mathrm{tail}(\bar{P}) = \mathrm{tail}(\bar{Q}), \tag{14}$$

for some sequences $\bar{t_1}, \ldots, \bar{t_m} > 0$, $\bar{a_1}, \ldots, \bar{a_m} \in Act$. Let $(\mathrm{tail}_H(P))^{-u}$ denote process $\mathrm{tail}_H(P)$ in which all time prefixes are decreased by u; analogously for $(\mathrm{tail}_H(P_0))^{-u}$. It is crucial now to observe that by the very definition of $\mathbb{DS}(\_,\, \_)$ we have

$$\bar{P} = \bar{P}'\|(\mathrm{tail}_H(P))^{-u} \quad \text{and} \quad \bar{Q} = \bar{Q}'\|(\mathrm{tail}_H(P))^{-u},$$

for some u such that $0 \leqslant u \leqslant H$ and some $(\bar{P}',\, \bar{Q}') \in \mathbb{DS}(P',\, Q')$, Moreover, if we put:

$$\bar{P}_0 := \bar{P}'\|(\mathrm{tail}_H(P_0))^{-u} \quad \text{and} \quad \bar{Q}_0 := \bar{Q}'\|(\mathrm{tail}_H(P_0))^{-u},$$

then we have $(\bar{P}_0,\, \bar{Q}_0) \in \mathbb{DS}(P'\|\mathrm{tail}_H(P_0),\, Q'\|\mathrm{tail}_H(P_0))$, by the very definition of $\mathbb{DS}(\_,\, \_)$ again. Now if we put $\bar{t_i} := t_i - u$ and $\bar{a_i} := a_i$, by (12) it is routine to check that (14) holds. □

6. Decision procedure

Our algorithm is composed of two semi-decision procedures, analogously as in the case of BPP. The TBPP processes are *finitely branching*, hence bisimulation inequivalence $\not\sim$ is semi-decidable. The semi-decision procedure is essentially the same as for BPP [6] and is based on the fact that $\sim$ is the intersection of a countable chain of decidable approximations. Furthermore, our development in Section 5 enables us to prove also semi-decidability of $\sim$ essentially in the same way as it was done for BPP. The semi-decision procedure for $\sim$ proposed below is motivated by the idea of Jančar [22] to use Presburger arithmetic.

For BPP, the crucial observation was that BPP processes can be seen as elements of the free commutative semigroup over *Const* with the semigroup operation $\_\|\_$, isomorphic to $\mathbb{N}^{Const}$ with vector addition $\_+\_$. The TBPP processes can be rather seen as elements of $\mathbb{N}^{\mathbb{N}\times Const}$. But due to Lemma 7, we can essentially restrict ourselves to only processes P with $\mathrm{maxclock}(P) \leqslant 2{\cdot}H$, i.e., to $\mathbb{N}^{\{0,\ldots,2\cdot H\}\times Const}$. This allows us to apply Theorem 2.

Definition 7. A *linear* subset of $\mathbb{N}^n$, $n > 0$, is a set of the form

$$v + span\{w_1, \ldots, w_k\} = \{v + n_1 w_i + \cdots + n_k w_k : n_1, \ldots, n_k \in \mathbb{N}\}.$$

i.e., each linear set is determined by a *base* $v \in \mathbb{N}^n$ and *periods* $\{w_1, \ldots, w_k\} \subset \mathbb{N}^n$. A *semi-linear* set is a finite union of linear sets.

Every binary relation over $\mathbb{N}^n$, i.e., a subset of $\mathbb{N}^n \times \mathbb{N}^n$, can be seen naturally as a subset of $\mathbb{N}^{2\cdot n}$. In the sequel we silently identify $\mathbb{N}^n \times \mathbb{N}^n$ and $\mathbb{N}^{2\cdot n}$.

Theorem 2 (*Eilenberg and Schuetzenberger [16], Hirshfeld [21], Jančar[22]*). *Each congruence in* $(\mathbb{N}^n, \_+\_), n>0$, *is semi-linear.*

By Theorem 2, $\sim$ restricted to the almost-identical pairs (P, Q) with $\mathrm{maxclock}(P)\leqslant 2{\cdot}H$ is semi-linear, as it is obviously a congruence w.r.t. $\_\|\_$. Hence we refine Lemma 7 as follows:

Corollary 1. *Let P and Q be almost-identical with* $\mathrm{maxclock}(P)\leqslant 2{\cdot}H$. *Then $P\sim Q$ iff P and Q are related by a semi-linear bounded bisimulation.*

Lemma 8. *For a semi-linear binary relation R over $\mathbb{N}^{\{0,\ldots,2\cdot H\}\times Const}$, given by a finite set of base-periods pairs, it is decidable whether R is a bounded bisimulation.*

Proof. Given R, we can effectively construct a closed formula in Presburger arithmetic ϕ_R such that ϕ_R is valid iff R is a bounded bisimulation. The validity problem for Presburger arithmetic is decidable (see for instance [27]), hence this gives a decision procedure required.

Let $Const = \{X_1, \ldots, X_N\}$. The formula ϕ_R involves several tuples of variables of the form $\{x_{t,i}\}_{0\leqslant t\leqslant 2\cdot H, 1\leqslant i\leqslant N}$, denoted in short $\bar{x}$. A valuation of each such tuple corresponds to a TBPP process.

The structure of ϕ_R follows directly the two points in Definition 6 (we omit the second clause in the first point of Definition 6, which is symmetric to the first clause):

$$\begin{aligned}\forall\, \bar{x}, \bar{y}.\ (\bar{x}, \bar{y}) \in R \Longrightarrow\ & (\bar{x}, \bar{y}) \text{ is an almost-identical pair}\\ \wedge\ & \left(\wedge_{r_x\in\Delta}\ \forall \bar{x'}.\ \bar{x} \xrightarrow{0,r_x} \bar{x'} \implies \exists \bar{y'}.\ \vee_{r_y\in\Delta \text{ labeled as } r_x}\ (\bar{y} \xrightarrow{0,r_y} \bar{y'} \wedge \mathbb{DS}(\bar{x'}, \bar{y'}) \subseteq R)\right)\\ \wedge\ & \left(\wedge_{r\in\Delta}\ \wedge_{1\leqslant t\leqslant H}\ \forall \bar{x'}.\ \bar{x} \xrightarrow{t,r} \bar{x'} \implies \exists \bar{y'}.\ (\bar{y} \xrightarrow{t,r} \bar{y'} \wedge (\bar{x'}, \bar{y'}) \in R)\right).\end{aligned}$$

We argue that all ingredients of ϕ_R are expressible in Presburger arithmetic. First, it is well-known that semi-linear sets are expressible in Presburger arithmetic [18]. Hence there exists a formula to denote "$(\bar{x}, \bar{y}) \in R$", with free variables $\bar{x}, \bar{y}$. Further, "$(\bar{x}, \bar{y})$ is an almost-identical pair" is easily expressible by a conjunction of $2{\cdot}H \times N$ equations. "$\mathbb{DS}(\bar{x'}, \bar{y'}) \subseteq R$" is expressible by a conjunction of $2{\cdot}H + 1$ formulas of the form "$\bar{x} \neq \bar{y} \Rightarrow (\bar{x}, \bar{y}) \in R$". Finally, let

$$r = (X_i \xrightarrow{a} 0 \rhd P_0 \| \ldots \| H \rhd P_H) \in \Delta,$$

for some $1\leqslant i\leqslant N$, where $P_u = X_1^{p_{u,1}} \| \ldots \| X_N^{p_{u,N}}$, $p_{u,j}\geqslant 0$ for $0\leqslant u\leqslant H$ and $1\leqslant j\leqslant N$. Now "$\bar{x} \xrightarrow{t,r} \bar{x'}$", $0\leqslant t\leqslant H$, is a shorthand for

$$x_{t,i} > 0\ \wedge\ x_{t,i} + p_{0,i} = x'_{t,i} + 1\ \wedge_{1\leqslant j\leqslant N,\ 0\leqslant u\leqslant H,\ (u,j)\neq(0,i)}\ x'_{t+u,j} = x_{t+u,j} + p_{u,j}. \qquad \square$$

Corollary 1 and Lemma 8 form the core of the semi-decidability proof. The semi-decision procedure for $P \sim Q$ consists of enumerating base-periods representations of all the semi-linear binary relations over $\mathbb{N}^{\{0,\ldots,2\cdot H\}\times Const}$ and checking whether any of them is a bounded bisimulation and contains the pair (P, Q). Hence we have proved:

Theorem 3. *Strong bisimilarity is decidable for TBPP.*

7. Bisimulation Game

It is possible to modify the rules of Bisimulation Game so that bounded bisimulations correspond directly to Defender's winning strategies. Instead, we consider in this section a much simpler modification of the original Bisimulation Game. The only modification is that we put a restriction on time-stamps: the time-stamp of each move is at least as big (late) as the biggest (latest) time-stamp in all proceeding moves, decreased by h, for a fixed non-negative integer h. Formally, the h-game is defined as follows. Arena contains now triples (P, Q, t), for processes P, Q and $t \in \mathbb{N}$; t is understood as the minimal value of time-stamp allowed in future moves. From (P, Q, t), Attacker is allowed to perform a transition from P or Q, say $P \xrightarrow{u,a} P'$, only if $u\geqslant t$. The Defender's response is the same as in the original

game, say $Q \xrightarrow{u,a} Q'$; and then the game continues from (P', Q', t'), where $t' = \max(t, u-h)$. We will write $P \sim_h Q$ if Defender has a winning strategy in h-game starting in configuration $(P, Q, 0)$.

It is routine to provide a corresponding notion of bisimulation. A family of binary relations $(R_t)_{t\in\mathbb{N}}$ over processes is a h-bisimulation if for each t, $(P, Q) \in R_t$, $a \in Act$ and $u \geqslant t$,

- if $P \xrightarrow{u,a} P'$ then $Q \xrightarrow{u,a} Q'$ for some Q' such that $(P', Q') \in R_{t'}$,
- if $Q \xrightarrow{u,a} Q'$ then $P \xrightarrow{u,a} P'$ for some P' such that $(P', Q') \in R_{t'}$,

where $t' = \max(t, u-h)$. It is easy to show

Lemma 9. *$P \sim_h Q$ if and only if $(P, Q) \in R_0$, for some h-bisimulation $(R_t)_{t\in\mathbb{N}}$.*

Proof. A h-bisimulation $(R_t)_{t\in\mathbb{N}}$, with $(P, Q) \in R_0$, provides a strategy for Defender, hence $P \sim_h Q$.

For the opposite direction, assume $P \sim_h Q$. Define the relation $\sim_h^t$, for any $t \in \mathbb{N}$, as follows: $P_1 \sim_h^t P_2$ iff Defender has a winning strategy in h-game starting in configuration (P_1, P_2, t). It is routine to check that $(\sim_h^t)_{t\in\mathbb{N}}$ is a h-bisimulation. And clearly $\sim_h = \sim_h^0$, hence $P \sim_h^0 Q$ as required. □

Apparently, h-game puts an additional restriction on Attacker only, compared to original Bisimulation Game. Hence, if $P \sim Q$ then $P \sim_h Q$, for any h. We will prove the opposite, for $h = H$:

Theorem 4. *Let P and Q be almost-identical and* $\mathrm{maxclock}(P) = \mathrm{maxclock}(Q) \leqslant 2 \cdot H$. *Then $P \sim Q$ if and only if $P \sim_H Q$.*

Proof. It is straightforward to show that for any h, $\sim_h$ is a congruence and that it enjoys the properties proved for $\sim$ in Lemmas 1 and 2 in Section 4—it turns out that the additional restriction on Attacker's moves does not violate compositionality and the decomposition properties. Then, by routine adaptations of the proofs one can show the following analogs of Lemmas 3 and 5 from Section 4:

Claim 1. *$P \sim_h Q$ iff $\mathbb{DS}(P, Q) \subseteq \sim_h$.*

Claim 2. *Whenever P and Q are almost-identical, $P \sim_h Q$ and* $\mathrm{tail}(P) \xrightarrow{u,a} P'$, *for $u \leqslant h$ then* $\mathrm{head}(P) \| P' \sim_h \mathrm{head}(Q) \| P'$.

A difference between Claim 2 and Lemma 5 is a restriction $u \leqslant h$; this guarantees $\max(0, u - h) = 0$.

Let relation R contain all almost-identical pairs (P, Q) such that $P \sim_H Q$ and

$$\mathrm{maxclock}(P) = \mathrm{maxclock}(Q) \leqslant 2 \cdot H.$$

We will show that R is a bounded bisimulation. Condition (ii) in Definition 6 in Section 5 follows immediately by Claim 2. For condition (i), consider any $(P, Q) \in R$ and a transition at $t = 0$ from P or Q, say $P \xrightarrow{0,a} P'$. Since $P \sim_H Q$, there is a Defender's response leading to a new pair $(P', Q') \in \sim_H$. And by Claim 1 we get that $\mathbb{DS}(P', Q') \subseteq \sim_H$, hence $\mathbb{DS}(P', Q') \subseteq R$ as required. □

The following is a conclusion from Theorem 4. If Attacker has a winning strategy in Bisimulation Game, in its original formulation from Section 2, then there exists also a winning strategy obeying the following restriction: the time-stamp of each move is at least as big (late) as the biggest (latest) time-stamp in all proceeding moves, decreased by H. Call an Attacker's strategy *well-timed* if the sequence of time-stamps is weakly monotonic (non-decreasing) in each play consistent with that strategy. In other words, well-timed Attacker's strategies are precisely strategies in 0-game. We have deduced that Attacker has always a strategy that is close to being well-timed. It is an interesting open question whether Attacker has always a well-timed winning strategy, assumed that she wins. We managed neither to prove this claim neither to disprove it by a counterexample. If proved to hold, the claim would lead to a significant simplification of the decision procedure described in Section 6.

8. Extensions

The proposed method of bisimilarity checking is quite robust and can be easily extended and adapted to other frameworks. Below we briefly sketch the proof of decidability of $\sim$ when communication is allowed and when well-timed semantics is considered instead of ill-timed one.

8.1. Communication

Assume that *Act* contains a distinguished silent action τ. Moreover, assume that for each $a \neq \tau$, there exists a complementary action $\bar{a} \in Act$ such that $\bar{\bar{a}} = a$. Two different synchronization rules have been proposed for timed processes, for $a \neq \tau$. The most restrictive rule, studied e.g. in [2],

$$\frac{P \xrightarrow{t,a} P' \quad Q \xrightarrow{t,\bar{a}} Q'}{P \| Q \xrightarrow{t,\tau} P' \| Q'} \tag{15}$$

prevents from any waiting, and two processes can synchronize only if they are ready to perform complementary actions at the same time (see [14] for a detailed discussion).

It is routine to verify that all the facts proved in Sections 4 and 5 are still valid; in particular, $\sim$ is still a congruence. Moreover, Lemma 8 from Section 6 still holds, since the Presburger formula constructed in the proof can be easily adapted for τ-moves arising from communication. Hence the decision procedure is exactly the same as in Section 6 and we conclude:

Theorem 5. *Strong bisimilarity is decidable for TBPP with rule* (15).

Now, we want to allow busy waiting when one of components is ready to execute an action before the other. Let us replace (15) by the following rule, corresponding to semantics studied in [20]:

$$\frac{X \xrightarrow{0,a} P \quad Y \xrightarrow{0,\bar{a}} Q \quad t = \max\{t_1, t_2\}}{t_1 \rhd X \| t_2 \rhd Y \xrightarrow{t,\tau} t \rhd (P \| Q)}. \tag{16}$$

Recall that processes are distinguished only up to structural congruence. This allows us, intuitively, to re-order parallel components to pick up the two communicating sub-processes. Moreover, the rule allows busy waiting *only* in the case of communication. When rule (16) is adopted, even Lemma 2 from Section 4 does not hold and it is not clear how to adapt our proof for this case. As a simple counterexample consider the following process definition:

$$A \xrightarrow{a} A, \quad A' \xrightarrow{a} A', \quad A' \xrightarrow{\bar{a}} A, \quad A'_1 \xrightarrow{\bar{a}} \mathbf{0}$$

and the following pair of bisimilar processes:

$$A \| A \| 1 \rhd (A'_1 \| A), \quad A \| A \| 1 \rhd A'. \tag{17}$$

Obviously $1 \rhd (A'_1 \| A)$ and $1 \rhd A'$, or equivalently $A'_1 \| A$ and A', are not equivalent, since only one of them can exhibit communication. On the other hand, to see that processes in (17) are equivalent, notice that both are capable of performing $\bar{a}$ exactly once, either in communication or separately. While in the right-hand process the communication can be derived in a unique way, in the other process there are two possibilities. In each case, after the communication both processes are equivalent to $A \| 1 \rhd A$, since $A \| A$, $A \| \mathbf{0}$ and A are equivalent.

Note that we did not use τ in rewriting rules, hence the counterexample is still valid when explicit use of τ is forbidden (like in CPP, cf. [8]).

8.2. Well-timed semantics

In well-timed semantics a transition $P \xrightarrow{t,a} P'$ is allowed exclusively at time $t = \mathrm{minclock}(P)$. This is even more restrictive for Attacker than 0-game in Section 7, as in 0-game, a transition $P \xrightarrow{u,a} P'$ may still be performed if

$u >$ minclock(P); and the only consequence is that all consecutive moves should have time-stamps greater or equal to u.

We lose now all the decomposition properties from Section 4 except Lemma 1. Hence we can consider only pairs (P, Q) with

$$\text{minclock}(P) = 0 = \text{minclock}(Q).$$

We put additionally minclock$(\mathbf{0}) = 0$, so that the empty process is not ruled out. Two such P, Q are bisimilar iff they are related by some *zero-bisimulation*, that is a binary relation R over processes such that whenever $(P, Q) \in R$, for each $a \in Act$,

- if $P \xrightarrow{0,a} P'$ then $Q \xrightarrow{0,a} Q'$ for some Q' such that minclock$(P') =$ minclock(Q') and $(\text{now}(P'), \text{now}(Q')) \in R$;
- if $Q \xrightarrow{0,a} Q'$ then $P \xrightarrow{0,a} P'$ for some P' such that minclock$(P') =$ minclock(Q') and $(\text{now}(P'), \text{now}(Q')) \in R$.

Zero-bisimulations should not be confused with h-bisimulations from Section 7. A zero-bisimulation gives rise to a zero-bisimulation, but the converse is not true.

Due to the bound H on maxclock(_) of all right-hand sides of rewrite rules, there always exists a zero-bisimulation containing only pairs (P, Q) with maxclock$(P) \leqslant H$ and maxclock$(Q) \leq H$, called a *bounded zero-bisimulation* below (maxclock(P) needs not be equal to maxclock(Q), even when P and Q are bisimilar). Strong bisimilarity is a congruence again, hence by Theorem 2 we derive:

Lemma 10. *Assume* minclock$(P) =$ minclock(Q). *$P \sim Q$ iff P and Q are related by a semi-linear bounded zero-bisimulation.*

Lemma 11. *For a semi-linear binary relation R over $\mathbb{N}^{\{0,\ldots,H\}\times Const}$, given by a finite set of base-periods pairs, it is decidable whether R is a bounded zero-bisimulation.*

Proof. Similarly as in Lemma 8, given R, we can effectively construct a formula in Presburger arithmetic ϕ_R such that ϕ_R is valid iff R is a bounded zero-bisimulation. □

Similarly as in Section 6, Lemmas 10 and 11 form a core of the semi-decision procedure: it consists of enumerating (base-periods representations of) all the semi-linear binary relations over $\mathbb{N}^{\{0,\ldots,H\}\times Const}$ and checking whether any of them is a bounded zero-bisimulation and contains the pair we check for equivalence. Hence we have proved:

Theorem 6. *Strong bisimilarity is decidable for TBPP with rule* (15), *under well-timed semantics.*

In the same vein one can show decidability even when busy waiting is allowed—recall that we were not able to achieve this in the ill-timed setting. As a conclusion, well-timed semantics is robustly easier that the ill-timed counterpart. We omit the details here. The crucial observation is that the bound H on maxclock(_) of relevant related pairs can be applied as before and that the well-timed transition relation, including the necessary busy waiting of parallel components, is easily expressible in Presburger arithmetic.

9. Positive-duration fragment

Let Δ be a pure *positive-duration* process definition, i.e., the right-hand side expressions in Δ conform to the syntax (2) under the restriction $t > 0$ (cf. Section 2). As usual we assume also that Δ is guarded. Instead of transforming Δ into standard form, in this section we prefer to work with the full syntax. However, a transition transforms a pure process into an impure one, because the delays $1 \rhd \_$, describing durations of actions performed, accumulate and therefore may appear not accompanied syntactically by an action prefix. Hence we need to extend the syntax of pure processes slightly. Let *semi-pure* processes be given by the syntax that extends (2) with:

$$P \quad ::= \ldots \mid 1 \rhd P.$$

For instance, $t \rhd P$ is always semi-pure if P is pure.

First, let us emphasize a decomposition property of performance equivalence in positive-duration fragment. In Lemma 13 we prove that *cancellation* holds, which allows to prove easily the decomposition result in the following Lemma 14. This extends a similar result of [5]. Moreover, our proof method is simpler and more direct.

Lemma 12 (*Time progress*). *For any semi-pure P and $t \geqslant 0$, there is no infinite sequence of transitions labeled by t starting in P*:

$$P \xrightarrow{t,a_1} P' \xrightarrow{t,a_2} P'' \xrightarrow{t,a_3} \cdots.$$

Proof. By structural induction over P, we give a finite upper bound $|P|$ on the number of consecutive transitions labeled by t. When $P = a.t' \rhd P'$, this bound is equal 1, i.e., P can have *at most* 1 transition labeled by t in sequence, $P \xrightarrow{t,a} t' \rhd P'$, since $t' > 0$. When $P = 1 \rhd P'$, we put $|1 \rhd P'| = |P'|$. Furthermore, $|P_1 + P_2| = max\{|P_1|, |P_2|\}$ and $|P_1 \| P_2| = |P_1| + |P_2|$. Finally, when $P = X$, $|X| = |E|$, where $(X \stackrel{\mathtt{def}}{=} E) \in \Delta$. □

Lemma 13 (*Cancellation*). *Let P_1, P_2, Q_1 and Q_2 be semi-pure. If $P_1 \| Q_1 \sim P_2 \| Q_2$ and $Q_1 \sim Q_2$ then $P_1 \sim P_2$.*

Proof. Assume $P_1 \| Q_1 \sim P_2 \| Q_2$ and $Q_1 \sim Q_2$. We can substitute Q_1 for Q_2 to obtain: $P_1 \| Q_1 \sim P_2 \| Q_1$. The crucial insight is that Defender has a "copy–paste" strategy in the game for $P_1 \sim P_2$.

Formally, to show $P_1 \sim P_2$ we shall demonstrate that the relation

$$R = \{(P_1, P_2) : P_1 \| Q \sim P_2 \| Q \text{ for some } Q\}$$

is a performance bisimulation. For each $(P_1, P_2) \in R$ we need to show that each transition of P_1 can be matched by a transition of P_2, and vice versa. So, assume $P_1 \xrightarrow{t,a} P_1'$. Hence $P_1 \| Q \xrightarrow{t,a} P_1' \| Q$, where Q is chosen so that $P_1 \| Q \sim P_2 \| Q$. Therefore, $P_2 \| Q$ has a matching transition. If the matching transition is performed in P_2, i.e., $P_2 \xrightarrow{t,a} P_2'$ and $P_1' \| Q \sim P_2' \| Q$, we are done. Otherwise, assume that the matching transition is performed in Q, i.e.,

$$Q \xrightarrow{t,a} Q' \tag{18}$$

and $P_1' \| Q \sim P_2 \| Q'$. But due to (18), process $P_1' \| Q$ has again a transition labeled by t and a, $P_1' \| Q \xrightarrow{t,a} P_1' \| Q'$. And again, if the response of $P_2 \| Q'$ is performed by P_2, we are done. Otherwise,

$$Q' \xrightarrow{t,a} Q'' \tag{19}$$

and $P_1' \| Q' \sim P_2 \| Q''$. Now a crucial observation is that we cannot continue forever in this way, with transitions (18), (19), and so on, since this would yield an infinite sequence of transitions labeled by t and a,

$$Q \xrightarrow{t,a} Q' \xrightarrow{t,a} Q'' \xrightarrow{t,a} \cdots$$

which is excluded by Lemma 12. Hence after some finite number of iterations, a response must be eventually performed in P_2. This completes the proof. □

Lemma 14 (*Decomposition*). *Let P_1, P_2, Q_1 and Q_2 be pure. If $P_1 \| t \rhd Q_1 \sim P_2 \| t \rhd Q_2$ and $t > 0$ then $P_1 \sim P_2$ and $Q_1 \sim Q_2$.*

Proof. To prove $Q_1 \sim Q_2$, it is enough to observe that for fixed P_1 and P_2, the relation $\{(Q_1, Q_2) : P_1 \| t \rhd Q_1 \sim P_2 \| t \rhd Q_2 \text{ for some } t \geqslant 1\}$ is a performance bisimulation. Then $P_1 \sim P_2$ follows immediately by Lemma 13. □

Lemma 14 will be used in the proof of Theorem 7. Recall that this strong decomposition property does not hold when zero-durations are allowed, cf. Example 2 in Section 4.

Lemma 15. *If P is pure and $P \xrightarrow{0,a} Q$ then $Q = t \rhd P' \| P''$, for some $t > 0$, and pure P', P''.*

(Recall that equality is understood up to the structural congruence, cf. Section 2, and that a pure and positive-duration Δ is considered.)

Proof. Routine, by structural induction w.r.t. the length of derivation of a transition. □

We will prove that performance equivalence coincides with *distributed bisimilarity* [11], the variant of bisimilarity taking into account distribution of processes. Distributed bisimilarity can be decided in polynomial time in BPP [25] and, due to the coincidence, the algorithm carries over to performance equivalence.

Distributed bisimilarity was introduced in [7], but here we follow [11]. Given a plain (untimed) BPP process definition Γ, consider the following SOS transition rules for BPP process expressions:

$$\frac{}{a.P \xrightarrow{a} [P, \mathbf{0}]} \qquad \frac{(X \stackrel{\mathtt{def}}{=} P) \in \Delta \qquad P \xrightarrow{a} [P', P'']}{X \xrightarrow{a} [P', P'']}$$
$$\frac{P \xrightarrow{a} [P', P'']}{P + Q \xrightarrow{a} [P', P'']} \qquad \frac{P \xrightarrow{a} [P', P'']}{P \| Q \xrightarrow{a} [P', P'' \| Q]}. \tag{20}$$

We write $P \xrightarrow{a} [P', P'']$ if this transition can be derived from the above rules. The rules reflect a view on a process as distributed in space. A transition $P \xrightarrow{a} [P', P'']$ gives rise to a *local* derivative P', which intuitively records a *location* at which the action is observed, and a *concurrent* derivative P'', recording the part of the process separated from the local component.

Two BPP processes P_1 and P_2 are *distributed bisimilar* w.r.t. Γ, denoted $P_1 \sim^{\mathrm{d}}_{\Gamma} P_2$, if they are related by some *distributed bisimulation* R, that is a binary relation over BPP process expressions such that whenever $(P, Q) \in R$, for each $a \in Act$,

- if $P \xrightarrow{a} [P', P'']$ then $Q \xrightarrow{a} [Q', Q'']$ for some Q', Q'' such that $(P', Q') \in R$ and $(P'', Q'') \in R$;
- if $Q \xrightarrow{a} [Q', Q'']$ then $P \xrightarrow{a} [P', P'']$ for some P', P'' such that $(P', Q') \in R$ and $(P'', Q'') \in R$.

We aim at relating $\sim$ and $\sim^{\mathrm{d}}_{\Gamma}$, for Γ derived naturally from Δ. Actions in Γ will be pairs (a, t), for an action a of Δ and a delay $t \rhd \_$ appearing in Δ. First, let $\mathrm{u}(P)$ (*untimed* P) be a plain BPP expression obtained from P by replacing each action-delay pair $a.t \rhd P$ with $(a, t).P$. Similarly, let $\Gamma = \mathrm{u}(\Delta)$ be a BPP process definition obtained by replacing each right-hand side P with $\mathrm{u}(P)$. In the sequel we write $\sim^{\mathrm{d}}$ instead of $\sim^{\mathrm{d}}_{\mathrm{u}(\Delta)}$. Note that given a BPP process P', there is precisely one P such that $\mathrm{u}(P) = P'$.

The following lemma states a close relationship between transitions of P and $\mathrm{u}(P)$.

Lemma 16. *Assume P is pure.*

(i) *If $P \xrightarrow{0,a} t \rhd P' \| P''$ then $\mathrm{u}(P) \xrightarrow{(a,t)} [\mathrm{u}(P'), \mathrm{u}(P'')]$.*

(ii) *If $\mathrm{u}(P) \xrightarrow{(a,t)} [Q', Q'']$ then $P \xrightarrow{0,a} t \rhd P' \| P''$, for the unique P' and P'' with $\mathrm{u}(P') = Q'$ and $\mathrm{u}(P'') = Q''$.*

Proof. By structural induction w.r.t. the length of derivation of a respective transition.

Case $P = \mathbf{0}$: Obvious.

Case $P = a.t \rhd P'$: Immediate, since the only transitions of P and $\mathrm{u}(P)$ are:

$$P \xrightarrow{0,a} t \rhd P' = t \rhd P' \| \mathbf{0} \quad \text{and} \quad \mathrm{u}(P) = (a, t).\mathrm{u}(P') \xrightarrow{(a,t)} [\mathrm{u}(P'), \mathbf{0}].$$

Case $P = P' + P''$: If lemma holds for P' and P'', directly from the SOS rules (1) and (20) we conclude that it also holds for $P' + P''$, as $\mathrm{u}(P' + P'') = \mathrm{u}(P') + \mathrm{u}(P'')$.

Case $P = P' \| P''$: Assume lemma holds for P' and P''. For (i), w.l.o.g. assume that an action a of $P' \| P''$ is performed in P', i.e., $P' \xrightarrow{0,a} t \rhd P'_1 \| P'_2$ and hence

$$P' \| P'' \xrightarrow{0,a} t \rhd P'_1 \| P'_2 \| P''.$$

By induction assumption we conclude $\mathrm{u}(P') \xrightarrow{(a,t)} [\mathrm{u}(P'_1), \mathrm{u}(P'_2)]$ an hence also

$$\mathrm{u}(P')\|\mathrm{u}(P'') \xrightarrow{a} [\mathrm{u}(P'_1), \mathrm{u}(P'_2)\|\mathrm{u}(P'')].$$

Since u(_) preserves parallel composition, we have $\mathrm{u}(P')\|\mathrm{u}(P'') = \mathrm{u}(P'\|P'')$ and $\mathrm{u}(P'_2)\|\mathrm{u}(P'') = \mathrm{u}(P'_2\|P'')$, and we get the desired transition:

$$\mathrm{u}(P'\|P'') \xrightarrow{a} [\mathrm{u}(P'_1), \mathrm{u}(P'_2\|P'')].$$

The proof of (ii) is similar. W.l.o.g. assume that an action of $\mathrm{u}(P'\|P'') = \mathrm{u}(P')\|\mathrm{u}(P'')$ is performed in $\mathrm{u}(P')$, i.e., $\mathrm{u}(P') \xrightarrow{(a,t)} [P_1, P_2]$ and hence

$$\mathrm{u}(P')\|\mathrm{u}(P'') \xrightarrow{a} [P_1, P_2\|\mathrm{u}(P'')].$$

Again by induction assumption we deduce that $P' \xrightarrow{0,a} t \rhd P'_1\|P'_2$ for some P'_1, P'_2 with $\mathrm{u}(P'_1) = P_1$ and $\mathrm{u}(P'_2) = P_2$. Hence, we get

$$P'\|P'' \xrightarrow{0,a} t \rhd P'_1\|P'_2\|P'',$$

with $\mathrm{u}(P'_2\|P'') = P_2\|\mathrm{u}(P'')$.

Case $P = X$: immediate.

This completes the proof. □

Theorem 7. *Assume P_1 and P_2 are pure. $P_1 \sim P_2$ iff* $\mathrm{u}(P_1) \sim^{\mathrm{d}} \mathrm{u}(P_2)$.

Proof. IF-IMPLICATION: We shall prove that the relation R defined as:

$$\{(t_1 \rhd P_1\|\ldots\|t_k \rhd P_k,\ t_1 \rhd Q_1\|\ldots\|t_k \rhd Q_k) : k \geqslant 0,\ \ \forall_{1\leqslant i\leqslant k}\ \mathrm{u}(P_i) \sim^{\mathrm{d}} \mathrm{u}(Q_i)\}$$

is a performance bisimulation; P_i and Q_i range over pure processes. Consider $(t_1 \rhd P_1\|\ldots\|t_k \rhd P_k, t_1 \rhd Q_1\|\ldots\|t_k \rhd Q_k)$ in R and let $t_1 \rhd P_1\|\ldots\|t_k \rhd P_k \xrightarrow{t,a} \bar{P}$. As all P_i and Q_i are pure, there is some i such that $t = t_i$ and this transition is performed in P_i, $P_i \xrightarrow{0,a}$. By Lemma 15 we conclude $P_i \xrightarrow{0,a} t \rhd P'\|P''$, for some pure P', P'' and $t > 0$; and

$$\bar{P} \ = \ t_1 \rhd P_1\|\ldots\|t_{i-1} \rhd P_{i-1}\|(t_i + t) \rhd P'\|t_i \rhd P''\|t_{i+1} \rhd P_{i+1}\|\ldots\|t_k \rhd P_k.$$

By Lemma 16(i) we have $\mathrm{u}(P_i) \xrightarrow{(a,t)} [\mathrm{u}(P'), \mathrm{u}(P'')]$. Since $\mathrm{u}(P_i) \sim^{\mathrm{d}} \mathrm{u}(Q_i)$, we know that there are some Q' and Q'' such that $\mathrm{u}(Q_i) \xrightarrow{(a,t)} [Q', Q'']$ and $\mathrm{u}(P') \sim^{\mathrm{d}} Q'$ and $\mathrm{u}(P'') \sim^{\mathrm{d}} Q''$. Now we apply Lemma 16(ii) to get $Q_i \xrightarrow{0,a} t \rhd S'\|S''$ with $\mathrm{u}(S') = Q'$ and $\mathrm{u}(S'') = Q''$. Hence

$$t_1 \rhd Q_1\|\ldots\|t_k \rhd Q_k \xrightarrow{t,a} \bar{Q} \quad \text{and} \quad (\bar{P}, \bar{Q}) \in R,$$

where

$$\bar{Q} \ = \ t_1 \rhd Q_1\|\ldots\|t_{i-1} \rhd Q_{i-1}\|(t_i + t) \rhd S'\|t_i \rhd S''\|t_{i+1} \rhd Q_{i+1}\|\ldots\|t_k \rhd Q_k.$$

ONLY-IF-IMPLICATION: We will show that

$$R = \{(\mathrm{u}(P), \mathrm{u}(Q)) : P \sim Q, \text{ and } P, Q \text{ are pure}\}$$

is a distributed bisimulation. Assume $P \sim Q$, and let $\mathrm{u}(P) \xrightarrow{(a,t)} [P'_1, P'_2]$. By Lemma 16(ii), $P \xrightarrow{0,a} t \rhd P_1\|P_2$ for some P_1 and P_2 with $\mathrm{u}(P_1) = P'_1$ and $\mathrm{u}(P_2) = P'_2$. Since $P \sim Q$, $Q \xrightarrow{0,a} t' \rhd Q_1\|Q_2$ and $t \rhd P_1\|P_2 \sim t' \rhd Q_1\|Q_2$. Again by Lemma 16, $\mathrm{u}(Q) \xrightarrow{(a,t')} [\mathrm{u}(Q_1), \mathrm{u}(Q_2)]$.

If $Q_2 \neq \mathbf{0}$ then $t \rhd P_1 \| P_2 \sim t' \rhd Q_1 \| Q_2$ implies $t = t'$, as P_1, P_2, Q_1 and Q_2 are all pure. Otherwise, if $Q_2 = \mathbf{0}$, t' may be assumed to be equal t by the time cancellation law for $\mathbf{0}$ (cf. Section 2). So we have $t \rhd P_1 \| P_2 \sim t \rhd Q_1 \| Q_2$. By Lemma 14 we have $P_1 \sim Q_1$ and $P_2 \sim Q_2$, hence both $(P_1', \mathrm{u}(Q_1)) = (\mathrm{u}(P_1), \mathrm{u}(Q_1))$ and $(P_2', \mathrm{u}(Q_2)) = (\mathrm{u}(P_2), \mathrm{u}(Q_2))$ are in R. This completes the proof. □

By Theorem 7, we obtain:

Theorem 8. *There exists a polynomial-time algorithm for performance equivalence in the positive-duration fragment.*

9.1. Communication

To express communication, the distributed semantics (20) is usually extended by an additional rule (cf. [11]):

$$\frac{P \xrightarrow{a} [P', P''] \quad Q \xrightarrow{\bar{a}} [Q', Q'']}{P \| Q \xrightarrow{\tau} [P' \| Q', P'' \| Q'']}.$$

This rule is analogous to (15) in Section 8. If we admit both rules, we are still able to prove the coincidence stated in Theorem 7, if we assume that actions a and $\bar{a}$ involved into a communication have always equal durations. We omit here most of the details, being routine, and sketch only an adaptation of the proof of cancellation (Lemma 13).

Recall that a crucial insight was a Defender's copy–paste strategy in the game for $P_1 \sim P_2$, based on the strategy for $P_1 \| Q \sim P_2 \| Q$. Let $P_1 \xrightarrow{t,\tau} P_1'$, which can be either a communication or an explicit τ-prefix. As long as the Defender's answer to $P_1 \| Q \xrightarrow{t,\tau} P_1' \| Q$ is performed solely in P_2 or in Q, the argument as before applies, independently whether the answer is a communication or explicit τ. (Lemma 12 holds still.) But it can happen that after a number of paste–copy moves, which made Q evolve into some $\bar{Q}$, the answer is a communication between P_2 and $\bar{Q}$, due to

$$P_2 \xrightarrow{t,a} P_2' \quad \text{and} \quad \bar{Q} \xrightarrow{t,\bar{a}} \bar{Q}',$$

for some action a, i.e.,

$$P_1' \| \bar{Q} \sim P_2' \| \bar{Q}'.$$

But now, if we focus on action $\bar{a}$, the situation is precisely as in Lemma 13. Imagine an Attacker's move $P_1' \| \bar{Q} \xrightarrow{t,\bar{a}} P_1' \| \bar{Q}'$. Again the same argument can be used to show that after another series of paste–copy moves, the answer to $\bar{a}$ must be finally in P_2', i.e., $P_2' \xrightarrow{t,a} P_2''$. Hence, we derived $P_1' \| \tilde{Q} \sim P_2'' \| \tilde{Q}$, for some $\tilde{Q}$, and $P_2 \xrightarrow{t,a} P_2' \xrightarrow{t,\bar{a}} P_2''$. And since we are in positive-duration fragment, we conclude that the two transitions performed in P_2 are *independent*, i.e., in two different components of P_2—otherwise the time-stamps could not be the same. Hence, $P_2 \xrightarrow{t,\tau} P_2''$, as needed.

Distributed bisimilarity is still decidable in polynomial time when communication is allowed—this is a topic of a forthcoming paper. Hence, as a conclusion, performance equivalence is polynomial as well.

10. Final remarks

We proved decidability of strong bisimilarity, also known as performance equivalence, for timed BPP. The decision procedure is based on decidability of validity problem for Presburger arithmetic. Nevertheless we believe that after the crucial development of Sections 4 and 5, also other proof methods known for plain BPP could be used here. For instance, the decision procedure could be also based on searching for a finite bisimulation base [6] or for a successful tableau [12,11]. In the latter case, we suppose that the method proposed in [28], being a generalization of the tableau method of [12], could be further generalized to capture our setting.

Exact complexity of the problem rests still unknown. Since our problem subsumes strong bisimilarity checking for BPP, the PSPACE lower bound of Srba [29] applies as well. While in the case of plain BPP the complexity of strong bisimilarity was recently finally established by Jančar [23], it is still open whether his method can be translated to TBPP.

Another interesting open question concerns simplification of strategies in the Bisimulation Game. Assumed that Attacker wins from a given pair of processes, does she always have a well-timed winning strategy (cf. remark at the end of Section 5)? If true, this would lead to simplification of the decision procedure. Moreover, it would be much more likely that the approach of [23] applies for TBPP as well.

In the positive-duration fragment, we have proved coincidence of performance equivalence with distributed bisimilarity. This completes the picture of coincidences between different non-interleaving equivalences on BPP.

References

[1] L. Aceto, History preserving, causal and mixed-ordering equivalence over stable event structures, Fund. Inform. 17 (1992) 319–331.
[2] L. Aceto, D. Murphy, Timing and causality in process algebra, Acta Inform. 33 (4) (1996) 317–350.
[3] J. Baeten, C. Middelburg, Process algebra with timing: real time and discrete time, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, Elevier, Amsterdam, 2001, pp. 627–684.
[4] M. Bednarczyk, Hereditary history preserving bisimulation or what is the power of the future perfect in program logics, Technical Report, Polish Academy of Sciences, Gdańsk, 1991.
[5] B. Bérard, A. Labroue, P. Schnoebelen, Verifying performance equivalence for timed basic parallel processes, in: Proc. FOSSACS'00, Lecture Notes in Computer Science, Vol. 1784, 2000, pp. 35–47.
[6] O. Burkart, D. Caucal, F. Moller, B. Steffen, Verification of infinite structures, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, Elevier, Amsterdam, 2001, pp. 545–623.
[7] I. Castellani, Bisimulations for concurrency, Ph.D. Thesis, University of Edinburg, 1988.
[8] I. Castellani, Process algebras with localities, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, 2001, pp. 945–1046.
[9] D. Caucal, Graphes canoniques des graphes algebraiques, Informatique Theoretique et Applications (RAIRO) 24 (4) (1990) 339–352.
[10] K. Čerāns, Decidability of bisimulation equivalence for parallel timer processes, in: Proc. CAV'92, Lecture Notes in Computer Science, Vol. 663, 1992.
[11] S. Christensen, Decidability and decomposition in process algebras, Ph.D. Thesis, Department of Computer Science, University of Edinburgh, UK, 1993.
[12] S. Christensen, Y. Hirshfeld, F. Moller, Bisimulation equivalence is decidable for basic parallel processes, in: Proc. CONCUR'93, Lecture Notes in Computer Science, Vol. 713, 1993, pp. 143–157.
[13] F. Corradini, On performance congruences for process algebras, Inform. and Comput. 145 (2) (1998) 191–230.
[14] F. Corradini, R. Gorrieri, M. Roccetti, Performance preorder and competitive equivalence, Acta Inform. 34 (11) (1997) 805–835.
[15] F. Corradini, M. Pistore, Specification and verification of lazy timed systems, in: Proc. MFCS'96, Lecture Notes in Computer Science, Vol. 1113, Springer, Berlin, 1996, pp. 279–290.
[16] S. Eilenberg, M. Schuetzenberger, Rational sets in commutative monoids, J. Algebra 13 (1969) 173–191.
[17] S. Fröschle, Decidability of plain and hereditary history-preserving bisimulation for BPP, in: Proc. EXPRESS'99, Electronic Notes in Theoretical Computer Science, Vol. 27, 1999.
[18] S. Ginsburg, E. Spanier, Semigroups Presburger formulas and languages, Pacific J. Math. 16 (2) (1966) 285–296.
[19] R.v. Glabbeek, U. Goltz, Equivalence notions for concurrent systems and refinement of actions, in: Proc. MFCS'89, Lecture Notes in Computer Science, Vol. 379, Springer, Berlin, 1989, pp. 237–248.
[20] R. Gorrieri, M. Roccetti, E. Stancampiano, A theory of processes with durational actions, Theoret. Comput. Sci. 140 (1) (1995) 73–94.
[21] Y. Hirshfeld, Congruences in commutative semigroups, LFCS report ECS-LFCS-94-291, Laboratory for Foundations of Computer Science, University of Edinburgh, 1994.
[22] P. Jančar, Decidability questions for bisimilarity of Petri nets and some related problems, in: Proc. STACS'94, Lecture Notes in Computer Science, Vol. 775, Springer, Berlin, 1994, pp. 581–592.
[23] P. Jančar, Bisimilarity of basic parallel processes is PSPACE-complete, in: Proc. LICS'03, 2003, pp. 218–227.
[24] S. Lasota, Decidability of strong bisimilarity for timed BPP, in: Proc. CONCUR'02, Lecture Notes in Computer Science, Vol. 2421, Springer, Berlin, 2002, pp. 562–578.
[25] S. Lasota, A polynomial-time algorithm for deciding true concurrency equivalences of basic parallel processes, in: Proc. MFCS'03, Lecture Notes in Computer Science, Vol. 2747, Springer, Berlin, 2003, pp. 521–530.
[26] F. Moller, C. Tofts, Relating processes with respect to speed, in: Proc. CONCUR'91, Lecture Notes in Computer Science, Vol. 527, Springer, Berlin, 1991, pp. 424–438.
[27] D. Oppen, A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger arithmetic, J. Comput. System Sci. 16 (1978) 323–332.
[28] J. Srba, Note on the tableau technique for commutative transition systems, in: Proc. FOSSACS'02, Lecture Notes in Computer Science, Vol. 2303, Springer, Berlin, 2002, pp. 387–401.
[29] J. Srba, Strong bisimilarity and regularity of basic parallel processes is PSPACE-hard, in: Proc. STACS'02, Lecture Notes in Computer Science, Vol. 2285, Springer, Berlin, 2002, pp. 535–546.