Note on the Tableau Technique for Commutative Transition Systems

Jiří Srba⋆

BRICS⋆⋆, Department of Computer Science, University of Aarhus,
Ny Munkegade bldg. 540, 8000 Aarhus C, Denmark
srba@brics.dk

Abstract. We define a class of transition systems called effective commutative transition systems (ECTS) and show, by generalising a tableau-based proof for BPP, that strong bisimilarity between any two states of such a transition system is decidable. It gives a general technique for extending decidability borders of strong bisimilarity for a wide class of infinite-state transition systems. This is demonstrated for several process formalisms, namely BPP process algebra, lossy BPP processes, BPP systems with interrupt and timed-arc BPP nets.

1 Introduction

Semantics of various formalisms for description of concurrent processes like process algebra, Petri nets, pushdown systems and many others is usually given in terms of labelled transition systems. This provides a common ground for studying such systems, and the usually considered problems as *model checking* and *equivalence checking* (see e.g. [8]) can be defined purely in terms of labelled transition systems. In this paper we focus on the equivalence checking problem and show a general approach for extending known decidability borders of strong bisimilarity for commutative-based process formalisms. In particular, we examine the class of transition systems generated by algebras with the operator of parallel composition and we discuss its extensions with lossiness, interrupt and with time features.

It is known that strong bisimilarity is undecidable for a typical representative of fully parallel models — Petri nets [15]. Nevertheless, in [9, 10] Christensen, Hirshfeld and Moller proved using a tableau technique, that bisimilarity is decidable for an important fragment of Petri nets called *communication free Petri nets*. The complexity of this algorithm is still open — no primitive recursive upper bound is known. PSPACE-hardness of the problem was recently shown in [23]. The class of transition systems definable by communication free Petri nets can be equivalently described in terms of process algebra with a commutative operator for parallel composition and recursion. It is this formalism, usually

⋆ The author is supported in part by the GACR, grant No. 201/00/0400.

⋆⋆ **B**asic **R**esearch in **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

called *Basic Parallel Processes* (BPP), that is used in the original tableau-based proof in [10]. For an overview on the tableau technique consult e.g. [16].

We abstract from the specific BPP syntax and generalise the proof for a class of transition systems called *effective commutative transition systems* (ECTS). We give six simple conditions on a transition system to be an ECTS and if all of them are satisfied, bisimilarity between any two states of the transition system is decidable. There is no need to know the syntactic description of the system. Moreover, the generalisation is achieved in several ways: (i) states can be tuples of bounded multisets of natural numbers and not only tuples of natural numbers, (ii) we do not insist on a specific computation of successors of a given state — any effectively computable and finite set of successors is acceptable, and (iii) an auxiliary equivalence relation on states is introduced in order to check invariants for pairs in a bisimulation relation.

Semantics of many formalisms can be defined as an ECTS and this yields immediately decidability of bisimilarity. We demonstrate this on four examples — BPP process algebra, lossy BPP processes, BPP systems with interrupt and timed-arc BPP nets — thus extending in several ways the known decidability border which lies somewhere between BPP systems and state-extended BPP systems (state-extended BPP systems are a strict subclass of Petri nets where bisimilarity is still undecidable [8, 16]).

Note: full version of this paper appears as [22].

2 General Method

Let $\mathbb{N}_0 = \{0, 1, \ldots\}$ be the set of natural numbers. A *multiset* of $\mathbb{N}_0$ is a function $M : \mathbb{N}_0 \to \mathbb{N}_0$. Let $i \in \mathbb{N}_0$, then $M(i)$ denotes the number of occurrences of i in the multiset M. The *empty multiset* $\emptyset$ is a function such that $\emptyset(i) = 0$ for all $i \in \mathbb{N}_0$. The *multiset union* of two multisets M_1 and M_2 is defined by $(M_1 \uplus M_2)(i) = M_1(i) + M_2(i)$ for all $i \in \mathbb{N}_0$. By $\mathcal{B}_\infty$ we denote the set of all multisets of $\mathbb{N}_0$. Let $m \in \mathbb{N}_0$. We define a set $\mathcal{B}_m$ of all multisets of $\{0, 1, \ldots, m\}$, i.e., $M \in \mathcal{B}_m$ iff $M \in \mathcal{B}_\infty$ and $M(i) = 0$ for all $i \in \mathbb{N}_0$ such that $i > m$. We call a multiset $M \in \mathcal{B}_\infty$ *finite* if there is some $m \in \mathbb{N}_0$ such that $M \in \mathcal{B}_m$. For finite multisets we sometimes use an alternative set-like notation: e.g. a multiset $\{0, 1, 1, 4, 4, 4\}$ is the same as a multiset M such that $M(0) = 1$, $M(1) = 2$, $M(4) = 3$ and $M(i) = 0$ for $i \in \mathbb{N}_0 \smallsetminus \{0, 1, 4\}$.

Let $M, N \in \mathcal{B}_m$. We write $M \prec_\ell N$ iff there is k, $0 \le k \le m$, such that $M(k) < N(k)$ and $M(i) = N(i)$ for all i, $0 \le i < k$. Let $M, N \in \mathcal{B}_m$ then $M \neq N$ implies that either $M \prec_\ell N$ or $N \prec_\ell M$. We write $M \preceq_c N$ iff $M(i) \le N(i)$ for every i, $1 \le i \le m$, i.e., iff there is $M' \in \mathcal{B}_m$ such that $N = M \uplus M'$.

Let $m, n \in \mathbb{N}_0$ and $n > 0$. We define a structure $S = (\mathcal{B}_m^n, \oplus, \emptyset^n)$ where $\mathcal{B}_m^n$ is a set of n-tuples of elements from $\mathcal{B}_m$. Let $\alpha = (M_1, M_2, \ldots, M_n) \in \mathcal{B}_m^n$ and $\beta = (N_1, N_2, \ldots, N_n) \in \mathcal{B}_m^n$, then $\alpha \oplus \beta = (M_1 \uplus N_1, M_2 \uplus N_2, \ldots, M_n \uplus N_n)$. Of course, $\alpha \oplus \beta \in \mathcal{B}_m^n$. The structure S is a commutative monoid. If $\alpha \in \mathcal{B}_m^n$ then α_i, $1 \le i \le n$, is the i'th coordinate of α. We introduce two orderings on $\mathcal{B}_m^n$.

Let $\alpha, \beta \in \mathcal{B}_m^n$, then $\alpha <_\ell \beta$ iff there is k, $1 \le k \le n$, such that $\alpha_k \prec_\ell \beta_k$ and $\alpha_i = \beta_i$ for every i, $1 \le i < k$; and $\alpha \le_c \beta$ iff $\alpha_i \preceq_c \beta_i$ for every i, $1 \le i \le n$.

Observe that $<_\ell$ is a well-founded ordering (there is no infinite sequence $\alpha_1, \alpha_2, \ldots$ such that $\alpha_1 >_\ell \alpha_2 >_\ell \ldots$) since $\prec_\ell$ is well-founded. Moreover for any $\alpha \neq \beta$ either $\alpha <_\ell \beta$ or $\beta <_\ell \alpha$. Also notice that $\alpha \le_c \beta$ iff there is $\alpha' \in \mathcal{B}_m^n$ such that $\beta = \alpha \oplus \alpha'$. We write $\alpha <_c \beta$ iff $\alpha \le_c \beta$ and $\alpha \neq \beta$. The following lemma is a simple generalisation of Dickson's Lemma [12].

Lemma 1. *Every infinite sequence from $\mathcal{B}_m^n$ has an infinite nondecreasing subsequence w.r.t. $\le_c$.*

A *labelled transition system* is a 4-tuple $(S, \mathcal{A}ct, \longrightarrow, \mathcal{E}qv)$ where S is a set of *states* (or *processes*), $\mathcal{A}ct$ is a set of *labels* (or *actions*), $\longrightarrow \subseteq S \times \mathcal{A}ct \times S$ is a *transition relation*, written $\alpha \xrightarrow{a} \beta$, for $(\alpha, a, \beta) \in \longrightarrow$, and $\mathcal{E}qv \subseteq S \times S$ is an *equivalence relation* on states.

Our definition of labelled transition systems is a generalisation of labelled transition systems with final states — see an overview paper [8]. Let $F \subseteq S$ be a set of final states. In order to recover the definition from [8] we define $(\alpha, \beta) \in \mathcal{E}qv$ iff $\alpha \in F$ and $\beta \in F$, or $\alpha \notin F$ and $\beta \notin F$.

Let $\alpha \in S$. We write $\alpha \not\longrightarrow$ whenever there is no $\beta \in S$ and $a \in \mathcal{A}ct$ such that $\alpha \xrightarrow{a} \beta$. As usual we extend the transition relation to the elements of $\mathcal{A}ct^*$. We define a *norm* of $\alpha \in S$ by $\mathcal{N}(\alpha) = \min\{|w| \mid w \in \mathcal{A}ct^* \text{ such that } \exists \beta \in S.\ \alpha \xrightarrow{w} \beta \not\longrightarrow\}$. By definition $\min \emptyset = \infty$.

Let $T = (S, \mathcal{A}ct, \longrightarrow, \mathcal{E}qv)$ be a labelled transition system. A binary relation $R \subseteq S \times S$ is a *bisimulation* iff whenever $(\alpha, \beta) \in R$ then for each $a \in \mathcal{A}ct$: if $\alpha \xrightarrow{a} \alpha'$ then $\exists \beta' \in S$ such that $\beta \xrightarrow{a} \beta'$ and $(\alpha', \beta') \in R$; if $\beta \xrightarrow{a} \beta'$ then $\exists \alpha' \in S$ such that $\alpha \xrightarrow{a} \alpha'$ and $(\alpha', \beta') \in R$; and $(\alpha, \beta) \in \mathcal{E}qv$.

States $\alpha, \beta \in S$ are *bisimulation equivalent* or *bisimilar* in a transition system T, written $\alpha \sim_T \beta$, iff $(\alpha, \beta) \in R$ for some bisimulation R. If T is clear from the context, we write only $\alpha \sim \beta$ instead of $\alpha \sim_T \beta$.

Remark 1. Sometimes the bisimilarity checking problem is formulated in this way: we are given a pair of labelled transition systems T_1 and T_2 with states α_1 from T_1 and α_2 from T_2, and the question $\alpha_1 \sim \alpha_2$ is asked. In this case, we can consider a disjoint union of T_1 and T_2 (i.e. the sets of states of T_1 and T_2 are disjoint) as a new transition system T and ask the question $\alpha_1 \sim_T \alpha_2$.

Let $(S, \mathcal{A}ct, \longrightarrow, \mathcal{E}qv)$ be a labelled transition system. The *stratified bisimulation relations* [19] $\sim_k \subseteq S \times S$ for $k \in \mathbb{N}_0$ are defined as follows:

- $\alpha \sim_0 \beta$ for all $\alpha, \beta \in S$ such that $(\alpha, \beta) \in \mathcal{E}qv$, i.e., $\sim_0 = \mathcal{E}qv$
- $\alpha \sim_{k+1} \beta$ iff for each $a \in \mathcal{A}ct$: if $\alpha \xrightarrow{a} \alpha'$ then $\exists \beta' \in S$ such that $\beta \xrightarrow{a} \beta'$ and $\alpha' \sim_k \beta'$; if $\beta \xrightarrow{a} \beta'$ then $\exists \alpha' \in S$ such that $\alpha \xrightarrow{a} \alpha'$ and $\alpha' \sim_k \beta'$; and $(\alpha, \beta) \in \mathcal{E}qv$.

Given a labelled transition system $T = (S, \mathcal{A}ct, \longrightarrow, \mathcal{E}qv)$ we define $\mathsf{next}(\alpha, a) = \{\beta \in S \mid \alpha \xrightarrow{a} \beta\}$ for $\alpha \in S$ and $a \in \mathcal{A}ct$. We also define $\mathsf{next}(\alpha, *) = \bigcup_{a \in \mathcal{A}ct} \mathsf{next}(\alpha, a)$. The system T is *image-finite* iff the set $\mathsf{next}(\alpha, a)$ is finite for every $\alpha \in S$ and $a \in \mathcal{A}ct$. The following lemma is a standard one.

Lemma 2. *Let $(S, \mathcal{A}ct, \longrightarrow, \mathcal{E}qv)$ be an image-finite labelled transition system and $\alpha, \beta \in S$. Then $\alpha \sim \beta$ iff $\alpha \sim_k \beta$ for all $k \in \mathbb{N}_0$.*

Definition 1 (Effective Commutative Transition System). *A labelled transition system $T = (S, \mathcal{A}ct, \longrightarrow, \mathcal{E}qv)$ is an* effective commutative transition system *(ECTS) iff there exist $n, m \in \mathbb{N}_0$, $n > 0$ such that the following conditions are satisfied:*

(1) $S = \mathcal{B}_m^n$,
(2) $\mathcal{A}ct$ is a finite set,
(3) given $\alpha, \beta \in S$ it is decidable whether $(\alpha, \beta) \in \mathcal{E}qv$,
(4) $\mathsf{next}(\alpha, a)$ is effectively constructible for every $\alpha \in \mathcal{B}_m^n$ and $a \in \mathcal{A}ct$,
(5) T is image-finite,
(6) if $\alpha \sim_k \beta$ then $(\alpha \oplus \gamma) \sim_k (\beta \oplus \gamma)$ for every $\alpha, \beta, \gamma \in \mathcal{B}_m^n$ and $k \in \mathbb{N}_0$.

Let us call the elements of $\mathcal{B}_m^n$ *processes.* Since any ECTS is image-finite (5), the fact that $\sim_k$ are congruences (6) together with Lemma 2 implies: *(6')* if $\alpha \sim \beta$ then $(\alpha \oplus \gamma) \sim (\beta \oplus \gamma)$ for every $\alpha, \beta, \gamma \in \mathcal{B}_m^n$.

Theorem 1. *Let $T = (\mathcal{B}_m^n, \mathcal{A}ct, \longrightarrow, \mathcal{E}qv)$ be an ECTS. Given $A, B \in \mathcal{B}_m^n$, it is decidable whether $A \sim B$.*

Proof. The proof is by tableau-technique and it is a generalisation of the tableau-based proof used by Christensen, Hirshfeld and Moller in order to demonstrate decidability of bisimilarity for BPP [9, 10].

A tableau for $(A, B) \in \mathcal{B}_m^{2n}$ is a maximal proof tree rooted with (A, B) and built according to the following rules. Let (α, β) be a node in the tree. A node (α, β) is either *terminal (leaf)* or *nonterminal.* The following nodes are terminal:

– (α, α) is a *successful leaf* for any $\alpha \in \mathcal{B}_m^n$ (note that always $(\alpha, \alpha) \in \mathcal{E}qv$),
– (α, β) is a *successful leaf* if $\mathsf{next}(\alpha, *) \cup \mathsf{next}(\beta, *) = \emptyset$ and $(\alpha, \beta) \in \mathcal{E}qv$,
– (α, β) is an *unsuccessful leaf* if for some $a \in \mathcal{A}ct$ it is the case that $\mathsf{next}(\alpha, a) \cup \mathsf{next}(\beta, a) \neq \emptyset$, and either $\mathsf{next}(\alpha, a) = \emptyset$ or $\mathsf{next}(\beta, a) = \emptyset$,
– (α, β) is an *unsuccessful leaf* if $(\alpha, \beta) \notin \mathcal{E}qv$.

We say that a node is an *ancestor* of (α, β) if it is on the path from the root to (α, β) and at least one application of the rule EXPAND (defined later) separates them. If (α, β) is not a leaf then we reduce it using the following RED rules as long as possible.

$$\mathrm{RED}_L \quad \frac{(\alpha, \beta)}{(\gamma \oplus \omega, \beta)} \quad \begin{array}{l}\text{if there is an ancestor } (\gamma, \delta) \text{ or } (\delta, \gamma) \text{ of } (\alpha, \beta) \text{ such that} \\ \gamma <_\ell \delta \text{ and } \alpha = \delta \oplus \omega \text{ for some } \omega \in \mathcal{B}_m^n\end{array}$$

$$\mathrm{RED}_R \quad \frac{(\alpha, \beta)}{(\alpha, \gamma \oplus \omega)} \quad \begin{array}{l}\text{if there is an ancestor } (\gamma, \delta) \text{ or } (\delta, \gamma) \text{ of } (\alpha, \beta) \text{ such that} \\ \gamma <_\ell \delta \text{ and } \beta = \delta \oplus \omega \text{ for some } \omega \in \mathcal{B}_m^n\end{array}$$

If no other reduction RED is applicable and the resulting node is not a leaf, we apply the rule EXPAND for a set of relations S_a, $a \in \mathcal{A}ct$, where $S_a \subseteq$

$\mathsf{next}(\alpha, a) \times \mathsf{next}(\beta, a)$ such that $\forall \alpha' \in \mathsf{next}(\alpha, a).\exists \beta' \in \mathsf{next}(\beta, a).\ (\alpha', \beta') \in S_a$ and $\forall \beta' \in \mathsf{next}(\beta, a).\exists \alpha' \in \mathsf{next}(\alpha, a).\ (\alpha', \beta') \in S_a$.

$$\text{EXPAND} \ \frac{(\alpha, \beta)}{\{(\alpha', \beta') \mid a \in \mathcal{A}ct \ \wedge \ (\alpha', \beta') \in S_a\}}$$

The set notation used in the rule EXPAND means that each element (α', β') in the conclusion of the rule becomes a new child in the proof tree. Now, we start again applying the RED-rules to every such child (which is not a leaf) as long as possible. Note that reduction rules are applicable to a node iff the node is not terminal (leaf).

Lemma 3. *Any tableau for (A, B) is finite and there are only finitely many tableaux.*

Proof. Observe that any tableau for (A, B) is finitely branching because of the assumption (5) and the condition that $\mathcal{A}ct$ is finite (2), which implies that for a given $a \in \mathcal{A}ct$ any relation S_a is finite and there are finitely many such relations. Should the tableau be infinite, there is an infinite branch, which gives an infinite sequence of vectors from $\mathcal{B}_m^{2n}$. Since the rules RED can be used only finitely many times in a sequence (they decrease the $<_\ell$ order, which is well founded), there must be an infinite subsequence of vectors on which the rule EXPAND was applied. Using Lemma 1, this sequence must contain an infinite nondecreasing subsequence $p_1 \leq_c p_2 \leq_c \ldots$. However, the rule EXPAND cannot be applied on p_2 since one of the rules RED is applicable. This is a contradiction.

Since there are only finitely many relations S_a for an $a \in \mathcal{A}ct$ available for the EXPAND rule and finitely many possibilities for an application of the RED rule, there are always finitely many possibilities how to extend already existing partial tableau. Suppose that there are infinitely many tableaux starting from (A, B). Then there must be a tableau for (A, B) with an infinite branch, which contradicts that every tableau is finite. □

We call a tableau for (A, B) *successful* if it is maximal (no further rules are applicable) and all its leaves are successful.

Lemma 4 (Completeness). *If $A \sim B$ then there is a successful tableau for (A, B).*

Proof. We construct a tableau from the root (A, B) such that every node (α, β) in the tableau satisfies $\alpha \sim \beta$. Hence this tableau cannot contain any unsuccessful leaf and it must be finite because of Lemma 3. Suppose that (α, β) is already a node in the tableau such that $\alpha \sim \beta$ and consider the rule RED_L applied on (α, β). We may assume that $\gamma \sim \delta$, which means using (6') that $(\gamma \oplus \omega) \sim (\delta \oplus \omega) = \alpha \sim \beta$. Hence $(\gamma \oplus \omega) \sim \beta$. Similarly for RED_R. From the definition of $\sim$ follows that the rule EXPAND is also forward sound, i.e., if $\alpha \sim \beta$ then we can choose for every $a \in \mathcal{A}ct$ a relation S_a such that $(\alpha', \beta') \in S_a$ implies that $\alpha' \sim \beta'$. □

Lemma 5 (Soundness). *If there is a successful tableau for* (A, B) *then* $A \sim B$*.*

Proof. For the sake of contradiction assume that there is a successful tableau for (A, B) and $A \not\sim B$. We show that we can construct a path from the root (A, B) to some leaf, such that for any pair (α, β) on this path $\alpha \not\sim \beta$.

If $A \not\sim B$ then using Lemma 2 there is a minimal k such that $A \not\sim_k B$. Notice that if $\alpha \not\sim_k \beta$ such that k is minimal and we apply the rule EXPAND, then at least one of its children (α', β') satisfies that $\alpha' \not\sim_{k-1} \beta'$. We choose such a child to extend our path from the root.

If we apply RED_L on (α, β) where $\alpha \not\sim_k \beta$ and k is minimal, then the corresponding ancestor (γ, δ) is separated by at least one application of EXPAND and so $\gamma \sim_k \delta$. This implies that $(\gamma \oplus \omega) \not\sim_k \beta$, otherwise using the assumption (6) we get that $\alpha = (\delta \oplus \omega) \sim_k (\gamma \oplus \omega) \sim_k \beta$, which is a contradiction with $\alpha \not\sim_k \beta$. The same is true for RED_R. Thus there must be a path from the root to some leaf such that for any pair (α, β) on this path $\alpha \not\sim \beta$. This is a contradiction with the fact that the path contains a successful leaf. □

We have proved that it is decidable whether $A \sim B$, since it is the case iff there is a successful tableau for (A, B). There are only finitely many tableaux and all of them are finite, moreover the conditions (3) and (4) ensure that they are effectively constructible. □

3 Applications

In this section we consider several specific classes of commutative transition systems. We study in particular BPP and lossy BPP processes, interrupt BPP systems and timed-arc BPP nets.

3.1 BPP and deadlock-sensitive BPP

The class of Basic Parallel Processes (BPP) [9] is a natural subclass of PA (Process Algebra) where only the operator of parallel composition is used. It is a well known fact that bisimilarity is decidable for BPP [9, 10]. We give the definition of BPP by means of process rewrite systems [17], which is more convenient for our purposes than the usual one by process equations used by Milner [19]. We remind the reader of the fact that these two definitions are equivalent in the sense that they define the same class of processes up to bisimilarity.

Let $\mathcal{A}ct$ and $\mathcal{V}ar$ be countable sets of *actions* and *process constants* such that $\mathcal{A}ct \cap \mathcal{V}ar = \emptyset$. We define a class of *process expressions* $\mathcal{E}^{\mathcal{V}ar}$ over $\mathcal{V}ar$ by the following abstract syntax $E ::= \epsilon \mid X \mid E \| E$, where ϵ is the *empty process* and X ranges over $\mathcal{V}ar$. The operator '$\|$' stands for a *parallel composition.* We do not distinguish between process expressions related by a *structural congruence* $\equiv \subseteq \mathcal{E}^{\mathcal{V}ar} \times \mathcal{E}^{\mathcal{V}ar}$, which is the smallest congruence over process expressions such that '$\|$' is associative and commutative, and 'ϵ' is a unit for '$\|$'.

A BPP *process rewrite system* (PRS) [17] is a finite set $\Delta \subseteq \mathcal{V}ar \times \mathcal{A}ct \times \mathcal{E}^{\mathcal{V}ar}$ of *rules*, written $X \xrightarrow{a} E$ for $(X, a, E) \in \Delta$. Let us denote the set of actions and

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

Fig. 1. SOS rules for BPP

process constants that appear in Δ as $\mathcal{A}ct(\Delta)$ resp. $\mathcal{V}ar(\Delta)$ (note that these sets are finite). A process rewrite system Δ determines a labelled transition system $T(\Delta) = (\mathcal{E}^{\mathcal{V}ar(\Delta)}/\equiv, \mathcal{A}ct(\Delta), \longrightarrow, \mathcal{E}qv)$ where states are $\equiv$-equivalence classes of process expressions over $\mathcal{V}ar(\Delta)$, $\mathcal{A}ct(\Delta)$ is the set of labels, the transition relation $\longrightarrow$ is the least relation satisfying the SOS rules in Figure 1 (recall that '$\|$' is commutative and in what follows we often abuse the notation and write only E instead of $[E]_\equiv$, i.e., the equivalence class represented by E). There are two possibilities for defining the equivalence relation $\mathcal{E}qv$. In the usual setting $\mathcal{E}qv = (\mathcal{E}^{\mathcal{V}ar(\Delta)}/\equiv) \times (\mathcal{E}^{\mathcal{V}ar(\Delta)}/\equiv)$ is the universal relation (thus it is in fact unused) and we call this class BPP. Another possibility is to define $\mathcal{E}qv$ by

$$\mathcal{E}qv = \{(E, F) \in (\mathcal{E}^{\mathcal{V}ar(\Delta)}/\equiv) \times (\mathcal{E}^{\mathcal{V}ar(\Delta)}/\equiv) \mid E = [\epsilon]_\equiv \iff F = [\epsilon]_\equiv\}.$$

We call this class *deadlock-sensitive* BPP. A study of strict (deadlock-sensitive) and nonstrict (deadlock-nonsensitive) bisimilarity for a sequential analogue of BPP called Basic Process Algebra (BPA) is provided in [21].

We show that given a BPP system Δ, we can interpret its semantics as a commutative transition system such that states are elements of $\mathcal{B}_{n-1} = \mathcal{B}^1_{n-1}$ where $n = |\mathcal{V}ar(\Delta)|$. Because of the structural congruence $\equiv$, any process expression E over $\mathcal{V}ar(\Delta)$ can be represented by a vector of n natural numbers. Suppose a fixed ordering on $\mathcal{V}ar(\Delta) = \{X_0, X_1, \dots, X_{n-1}\}$. Then the corresponding vector contains on i'th coordinate the number of occurrences of the variable X_i in E. Formally, we define a mapping $\phi : \mathcal{E}^{\mathcal{V}ar(\Delta)} \to \mathcal{B}^1_{n-1}$ by

$$\begin{aligned} \phi(\epsilon) &= \emptyset \\ \phi(X_i) &= M \text{ such that } M(i) = 1 \text{ and } M(j) = 0 \text{ for } j \neq i \\ \phi(E_1 \| E_2) &= \phi(E_1) \oplus \phi(E_2). \end{aligned}$$

The following proposition is an easy observation.

Proposition 1. *Let $E, F \in \mathcal{E}^{\mathcal{V}ar(\Delta)}$. Then $E \equiv F$ if and only if $\phi(E) = \phi(F)$.*

Hence any rule $(X \xrightarrow{a} E) \in \Delta$ can be represented by $\phi(X) \xrightarrow{a} \phi(E)$. The system Δ, where $n = |\mathcal{V}ar(\Delta)|$, generates a commutative labelled transition system $T^c(\Delta) = (\mathcal{B}^1_{n-1}, \mathcal{A}ct(\Delta), \longrightarrow, \mathcal{E}qv)$, where $\alpha \xrightarrow{a} \beta$ iff there exists a rule $(X \xrightarrow{a} E) \in \Delta$ such that $\alpha = \phi(X) \oplus \omega$ and $\beta = \phi(E) \oplus \omega$ for some $\omega \in \mathcal{B}^1_{n-1}$. The relation $\mathcal{E}qv$ for BPP and deadlock-sensitive BPP is defined in the same fashion as above.

Example 1. Let $\Delta = \{X_0 \xrightarrow{a} X_0 \| X_1 \| X_2 \| X_1,\ X_0 \xrightarrow{a} \epsilon,\ X_1 \xrightarrow{b} \epsilon,\ X_2 \xrightarrow{c} \epsilon\}$. Then $n = 3$ and e.g. $\phi(X_0) = \{0\}$, $\phi(X_0 \| X_1 \| X_2 \| X_1) = \{0, 1, 1, 2\}$ and $\phi(\epsilon) = \emptyset$.

$$\frac{(X \xrightarrow{a} E) \in \Delta}{X \xrightarrow{a} E} \qquad \frac{E \xrightarrow{a} E'}{E\|F \xrightarrow{a} E'\|F} \qquad \frac{}{E \xrightarrow{drop} F} \text{ if } \exists F' \neq \epsilon \text{ s.t. } E = F\|F'$$

Fig. 2. SOS rules for lossy BPP

A sequence of transitions $X_0 \xrightarrow{a} X_0\|X_1\|X_2\|X_1 \xrightarrow{a} X_0\|X_1\|X_2\|X_1\|X_1\|X_2\|X_1 \xrightarrow{b} X_0\|X_2\|X_1\|X_1\|X_2\|X_1 \xrightarrow{c} X_0\|X_1\|X_1\|X_2\|X_1$ has a straightforward analogue in $\mathcal{B}^1_2$: $\{0\} \xrightarrow{a} \{0,1,1,2\} \xrightarrow{a} \{0,1,1,1,1,2,2\} \xrightarrow{b} \{0,1,1,1,2,2\} \xrightarrow{c} \{0,1,1,1,2\}$.

Obviously, $T(\Delta)$ and $T^c(\Delta)$ are isomorphic labelled transition systems.

Theorem 2. *Given a BPP*[1] *process rewrite system Δ (or a deadlock-sensitive BPP process rewrite system Δ) and a pair of processes $P_1, P_2 \in \mathcal{E}^{\mathcal{V}\mathrm{ar}(\Delta)}/\equiv$, it is decidable whether $P_1 \sim_{T(\Delta)} P_2$.*

Proof. By Theorem 1 and the fact that $T^c(\Delta)$ defined above is an ECTS. $\square$

3.2 Lossy BPP

The notion of unreliability, in particular lossiness, has been intensively studied with a number of interesting results. Let us mention e.g. *Lossy Channel Systems* [1] and *Lossy Vector Addition Systems* [7, 18]. Lossy BPP systems were studied in [18] in the context of model checking problems. In lossy BPP we allow process constants disappear spontaneously at any time. We give a formal definition of lossy BPP systems first.

A *lossy* BPP *process rewrite system* is a finite set $\Delta \subseteq \mathcal{V}ar \times \mathcal{A}ct \times \mathcal{E}^{\mathcal{V}ar}$ of *rules*, written $X \xrightarrow{a} E$ for $(X, a, E) \in \Delta$. A process rewrite system Δ determines a labelled transition system $T(\Delta) = (\mathcal{E}^{\mathcal{V}\mathrm{ar}(\Delta)}/\equiv, \mathcal{A}ct(\Delta) \cup \{drop\}, \longrightarrow, \mathcal{E}qv)$ where states are $\equiv$-equivalence classes of process expressions over $\mathcal{V}ar(\Delta)$, $\mathcal{A}ct(\Delta) \cup \{drop\}$ is the set of labels with a distinguished label $drop \notin \mathcal{A}ct(\Delta)$ modelling lossiness, the transition relation $\longrightarrow$ is defined by the SOS rules in Figure 2 (we again abuse the notation and write only E instead of $[E]_\equiv$) and $\mathcal{E}qv$ for lossy BPP can be defined as in the case of BPP — deadlock sensitive or deadlock nonsensitive.

Example 2. Let $\Delta = \{X_0 \xrightarrow{a} X_0\|X_0,\ X_0 \xrightarrow{b} \epsilon\}$. Then $X_0 \longrightarrow^* X_0^k$ for any $k \in \mathbb{N}_0$ where $X_0^0 = \epsilon$ and $X_0^{k+1} = X_0\|X_0^k$. Also, $X_0^k \xrightarrow{drop} X_0^{k'}$ for any k', $0 \le k' < k$, in particular, $X_0^k \xrightarrow{drop} \epsilon$ and $\epsilon \not\longrightarrow$. This means that any reachable state in $T(\Delta)$ has norm at most 1. Moreover, $X_0^k \not\sim_{T(\Delta)} X_0^{k'}$ for any $k \neq k'$. Hence there cannot be any BPP process bisimilar to X_0 (there are only finitely many nonbisimilar

[1] For BPP this is already proved in [9, 10]. We repeat the theorem in order to demonstrate that our technique is general enough to cover already known results.

BPP states of norm less or equal to 1). On the other hand this property in general disallows to find a bisimilar lossy BPP process for a given BPP process. Thus the classes BPP and lossy BPP are, as expected, incomparable w.r.t. bisimilarity.

We are now ready to define semantics of lossy BPP in terms of commutative transition systems, similarly as for BPP. Let Δ be a lossy BPP system, where $n = |\mathcal{V}ar(\Delta)|$. By $T^c(\Delta) = (\mathcal{B}^1_{n-1}, \mathcal{A}ct(\Delta) \cup \{drop\}, \longrightarrow, \mathcal{E}qv)$ we denote a commutative transition system, where $\alpha \xrightarrow{a} \beta$ iff either (i) there is a rule $(X \xrightarrow{a} E) \in \Delta$ such that $\alpha = \phi(X) \oplus \omega$ and $\beta = \phi(E) \oplus \omega$ for some $\omega \in \mathcal{B}^1_{n-1}$, or (ii) $\beta <_c \alpha$ and $a = drop$. The relation $\mathcal{E}qv$ for lossy BPP is defined in the same fashion as mentioned above (deadlock sensitive or deadlock nonsensitive).

Obviously, $T(\Delta)$ and $T^c(\Delta)$ are isomorphic labelled transition systems. This implies the following decidability theorem for lossy BPP systems.

Theorem 3. *Given a lossy BPP process rewrite system Δ (either deadlock sensitive or deadlock nonsensitive) and a pair of processes $P_1, P_2 \in \mathcal{E}^{\mathcal{V}\mathrm{ar}(\Delta)}/\!\equiv$, it is decidable whether $P_1 \sim_{T(\Delta)} P_2$.*

Proof. By showing that $T^c(\Delta)$ is an ECTS and then by using Theorem 1 and the isomorphism between $T(\Delta)$ and $T^c(\Delta)$. Details can be found in [22]. □

3.3 Interrupt BPP

In this subsection we investigate *mode transfer operators* in BPP process algebra, in particular the interrupt operator. Quoting [3]: "A useful feature in programming languages and specification languages is the ability to denote mode switches. In particular, most languages have means to describe the disrupt and interrupt of the normal execution of a system." Various mode transfer operators were considered in the literature [2, 3, 5, 11, 13]. We define interrupt BPP systems that extend the pure BPP systems with an interrupt vector and a mechanism for handling the interrupt. The motivation is that every state is annotated with a set of allowed interrupts and if no interrupt appears, a normal execution of the process is performed. At any time an interrupt can be raised by performing the action *int*. A normal execution of the process is interrupted and the raised interrupt is handled. During this all interrupts are disallowed. After the interrupt is finished, the action *iret* is performed and a normal execution of the interrupted process continues.

Formally, an *interrupt* BPP *process rewrite system* Δ is a pair (Δ_1, Δ_2) where Δ_1 is a finite set $\Delta_1 \subseteq \mathcal{V}ar \times \mathcal{A}ct \times \mathcal{E}^{\mathcal{V}ar} \times 2^{\mathcal{V}\mathrm{ar}(\Delta_2)}$ and Δ_2 is a BPP system. We write $(X \xrightarrow{a} E, enable)$ for $(X, a, E, enable) \in \Delta_1$. By $\mathcal{V}ar(\Delta_1)$ we denote the set of variables that occur in the first and the third component of Δ_1. A process rewrite system $\Delta = (\Delta_1, \Delta_2)$ determines a labelled transition system $T(\Delta) = \big((\mathcal{E}^{\mathcal{V}\mathrm{ar}(\Delta_1)}/\!\equiv) \times 2^{\mathcal{V}\mathrm{ar}(\Delta_2)} \times \{0,1\} \times (\mathcal{E}^{\mathcal{V}\mathrm{ar}(\Delta_2)}/\!\equiv), \mathcal{A}ct(\Delta_1) \cup \mathcal{A}ct(\Delta_2) \cup \{int, iret\}, \longrightarrow, \mathcal{E}qv^u\big)$ where states are 4-tuples (E_1, IV, IF, E_2) such that E_1 is a BPP process, IV is an *interrupt vector*, IF is an *interrupt flag* (0 means normal execution and 1 means interrupt call) and E_2 is ϵ if $IF = 0$ or

$$\frac{(X \xrightarrow{a} E_1, enable) \in \Delta_1}{(X, IV, 0, \epsilon) \xrightarrow{a} (E_1, IV \cup enable, 0, \epsilon)}$$

$$\frac{(E_1, IV, 0, \epsilon) \xrightarrow{a} (E_1', IV', 0, \epsilon)}{(E_1 \| F_1, IV, 0, \epsilon) \xrightarrow{a} (E_1' \| F_1, IV', 0, \epsilon)}$$

$$\frac{X \in IV}{(E_1, IV, 0, \epsilon) \xrightarrow{int} (E_1, IV, 1, X)} \qquad \frac{}{(E_1, IV, 1, \epsilon) \xrightarrow{iret} (E_1, IV, 0, \epsilon)}$$

$$\frac{(X \xrightarrow{a} E_2) \in \Delta_2}{(E_1, IV, 1, X) \xrightarrow{a} (E_1, IV, 1, E_2)} \qquad \frac{(E_1, IV, 1, E_2) \xrightarrow{a} (E_1, IV, 1, E_2')}{(E_1, IV, 1, E_2 \| F_2) \xrightarrow{a} (E_1, IV, 1, E_2' \| F_2)}$$

Fig. 3. SOS rules for interrupt BPP

it contains the interrupt handling process in the case $IF = 1$. We assume that $int, iret \notin \mathcal{A}ct(\Delta_1) \cup \mathcal{A}ct(\Delta_2)$. The SOS rules for $\longrightarrow$ are defined in Figure 3 (E again represents $[E]_\equiv$ and '$\|$' is commutative) and for the sake of simplicity let us assume that $\mathcal{E}qv^u$ is the universal relation on states.

Example 3. Let $\Delta = \Big(\big\{(X_0 \xrightarrow{a} X_0\|X_0, \{Y_0\}),\ (X_0 \xrightarrow{b} \epsilon, \{Y_1\})\big\},\ \big\{Y_0 \xrightarrow{c} \epsilon,\ Y_1 \xrightarrow{d} Y_1\big\}\Big)$. Consider an initial state $(X_0, \emptyset, 0, \epsilon)$. Then the following sequence of transitions is possible in $T(\Delta)$: $(X_0, \emptyset, 0, \epsilon) \xrightarrow{a} (X_0\|X_0, \{Y_0\}, 0, \epsilon) \xrightarrow{b} (X_0, \{Y_0, Y_1\}, 0, \epsilon) \xrightarrow{int} (X_0, \{Y_0, Y_1\}, 1, Y_0) \xrightarrow{c} (X_0, \{Y_0, Y_1\}, 1, \epsilon) \xrightarrow{iret} (X_0, \{Y_0, Y_1\}, 0, \epsilon) \xrightarrow{int} (X_0, \{Y_0, Y_1\}, 1, Y_1) \xrightarrow{d} (X_0, \{Y_0, Y_1\}, 1, Y_1) \xrightarrow{d} (X_0, \{Y_0, Y_1\}, 1, Y_1) \xrightarrow{d} \cdots$. It is an easy observation that there is no BPP process bisimilar to the initial state $(X_0, \emptyset, 0, \epsilon)$ of $T(\Delta)$ — we use similar arguments as in Example 2.

We remind the reader of the fact that for any BPP system we can find a bisimilar interrupt BPP system simply by disallowing interrupts at all — we define $enable = \emptyset$ in every rule of the BPP system. Hence the class of interrupt BPP is strictly more expressive (w.r.t. bisimilarity) than the class of BPP.

We demonstrate now, how to give an alternative semantics in terms of a commutative transition system T^c. The idea is that the normal process execution is simulated one-to-one in T^c and the interrupt calls are checked using the relation $\mathcal{E}qv$ — thus there are no actions int and $iret$. Let $\Delta = (\Delta_1, \Delta_2)$ be an interrupt BPP system such that $\mathcal{V}ar(\Delta_1) = \{X_0, \ldots, X_{n_1-1}\}$ and $\mathcal{V}ar(\Delta_2) = \{Y_0, \ldots, Y_{n_2-1}\}$. In what follows we denote by $T(\Delta_2)$ the deadlock sensitive transition system generated by the BPP process Δ_2. Since bisimilarity in $T(\Delta_2)$ is decidable (Theorem 2), we may assume w.l.o.g. that $Y_i \not\sim_{T(\Delta_2)} Y_j$ for all i, j such that $0 \le i < j \le n_2 - 1$. Let $n = \max\{n_1 - 1, n_2 - 1\}$.

Let $T^c(\Delta) = (\mathcal{B}_n^2, \mathcal{A}ct(\Delta_1) \cup \mathcal{A}ct(\Delta_2), \longrightarrow, \mathcal{E}qv)$. The intuition is that in the first component of a state $(M, N) \in \mathcal{B}_n^2$ we remember a BPP expression of normal process execution and in the second component we remember an interrupt vector IV in the following sense: $N(i) = 0$ if $Y_i \notin IV$, and $N(i) > 0$ if $Y_i \in IV$. For

$\alpha = (M, N) \in \mathcal{B}_n^2$, let $IV(\alpha) = \{Y_i \mid 0 \leq i \leq n_2 - 1 \ \wedge \ N(i) > 0\}$ and let $cut(\alpha) = (M, N') \in \mathcal{B}_n^2$ such that

$$N'(i) = \begin{cases} 0 & \text{if } N(i) = 0 \\ 1 & \text{if } N(i) > 0 \end{cases}$$

for all $i \in \mathbb{N}_0$. We define $\alpha = (M, N) \xrightarrow{a} (M', N') = \alpha'$ iff $(E, IV(\alpha), 0, \epsilon) \xrightarrow{a} (E', IV(\alpha'), 0, \epsilon)$ such that $a \notin \{int, iret\}$, $\phi(E) = M$, $\phi(E') = M'$, and $cut(\alpha') = \alpha'$. The last condition ($cut(\alpha') = \alpha'$) ensures that $T^c(\Delta)$ becomes image-finite. Finally we define $\mathcal{E}qv$ as such a relation that for states $\alpha, \beta \in \mathcal{B}_n^2$: $(\alpha, \beta) \in \mathcal{E}qv$ iff $IV(\alpha) = IV(\beta)$.

The following property is an immediate consequence of the definition.

Property 1. Let $\alpha \in \mathcal{B}_n^2$. Then $\alpha \sim_{T^c(\Delta)} cut(\alpha)$.

Proposition 2. *Let $\Delta = (\Delta_1, \Delta_2)$ be an interrupt BPP system. Then it holds that $(E, \emptyset, 0, \epsilon) \sim_{T(\Delta)} (F, \emptyset, 0, \epsilon)$ iff $(\phi(E), \emptyset) \sim_{T^c(\Delta)} (\phi(F), \emptyset)$ for any $E, F \in \mathcal{E}^{\mathcal{V}ar(\Delta_1)}$.*

Proof. It is obvious that any transition under a where $a \notin \{int, iret\}$ in $T(\Delta)$ can be simulated naturally in the system $T^c(\Delta)$ and vice versa. An interrupt call in $T(\Delta)$ is checked using the relation $\mathcal{E}qv$ and whenever $(\alpha, \beta) \notin \mathcal{E}qv$ in $T^c(\Delta)$ then we can distinguish the corresponding states in $T(\Delta)$ by an appropriate interrupt call. □

Theorem 4. *Given an interrupt BPP process rewrite system Δ and a pair of processes $(E, \emptyset, 0, \epsilon)$, $(F, \emptyset, 0, \epsilon)$ in $T(\Delta)$, it is decidable whether $(E, \emptyset, 0, \epsilon) \sim_{T(\Delta)} (F, \emptyset, 0, \epsilon)$.*

Proof. By Proposition 2 it is enough to show that $T^c(\Delta)$ is an ECTS and then we use Theorem 1. Details can be found in [22]. □

Remark 2. We used BPP processes for interrupt handling (the system Δ_2). In fact, any process algebra where bisimilarity is decidable can be used.

3.4 Timed-Arc BPP

In this subsection we establish decidability of bisimilarity for a timed extension of BPP systems, called timed-arc BPP. It is worth mentioning another positive decidability result for timed BPP. The authors in [4] show that performance equivalence (a version of timed bisimilarity) is decidable in a polynomial time for BPP processes where actions have a certain time duration. However, their definition of timed BPP does not allow to interpret ordinary BPP systems as timed ones since a duration of an action cannot be equal to 0 and must be strictly positive. We define timed-arc BPP as a natural subclass of timed-arc Petri nets where time (age) is associated to tokens and transitions are labelled by time intervals, which restrict the age of tokens available for firing a transition —

see e.g. [6, 14]. Our definition implies that timed-arc BPP are a strict extension (w.r.t. bisimilarity) of ordinary BPP systems, as it is demonstrated later.

First, we introduce *labelled timed-arc Petri nets*, following definitions from [20] and then we define timed-arc BPP as its subclass where each transition has exactly one input place. A *labelled timed-arc Petri net* (LTAPN) is a tuple $N = (P, T, F, c, L, \lambda, \Sigma)$, where P is a finite set of *places*, T is a finite set of *transitions* such that $T \cap P = \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ is a *flow relation*, $c : F|_{P\times T} \to \mathbb{N}_0 \times (\mathbb{N}_0 \cup \{\infty\})$ is a *time constraint* on transitions such that for each arc $(p,t) \in F$ holds that $t_1 \le t_2$ where $c(p,t) = (t_1, t_2)$, L is a finite set of *labels*, $\lambda : T \to L$ is *a labelling function*, and $\Sigma \subseteq \mathbb{N}_0$ is a recursive set of allowed *time-elapsing steps*.

Let $x \in \mathbb{N}_0$ and $c(p,t) = (t_1, t_2)$. We write $x \in c(p,t)$ whenever $t_1 \le x \le t_2$. We also define ${}^\bullet t = \{p \mid (p,t) \in F\}$ and $t^\bullet = \{p \mid (t,p) \in F\}$. A *marking* M on N is a function $M\colon P \to \mathcal{B}$ where $\mathcal{B}$ denotes the set of all finite multisets on $\mathbb{N}_0$. Each place is thus assigned a certain number of tokens, and each token is annotated with a natural number (*age*). Let $x \in \mathcal{B}$ and $a \in \mathbb{N}_0$. We define $x \mathbin{<\!\!+} a$ such that we add the value a to every element of x, i.e., $x \mathbin{<\!\!+} a = \{b + a \mid b \in x\}$.

Let us now define the dynamics of LTAPNs. We introduce two types of transition rules: *firing* of a transition and *time-elapsing*. Let $N = (P, T, F, c, L, \lambda, \Sigma)$ be a LTAPN, M a marking and $t \in T$. We say that t is *enabled* by M iff $\forall p \in {}^\bullet t.\ \exists x \in M(p).\ x \in c(p,t)$. If t is enabled by M then it can be *fired*, producing a marking M' (written $M[t\rangle M'$) such that $\forall p \in P.\ M'(p) = \Big(M(p) \smallsetminus C^-(p,t)\Big) \cup C^+(t,p)$ where C^- and C^+ are chosen to satisfy the following equations (note that there may be more possibilities and that all the operations are on multisets):

$$C^-(p,t) = \begin{cases} \{x\} & \text{such that } x \in M(p) \text{ and } x \in c(p,t) \quad \text{if } p \in {}^\bullet t \\ \emptyset & \text{otherwise} \end{cases}$$

$$C^+(t,p) = \begin{cases} \{0\} & \text{if } p \in t^\bullet \\ \emptyset & \text{otherwise.} \end{cases}$$

Note that the tokens added to places $t^\bullet$ are of age 0. We define also *time-elapsing* transitions τ_k, $k \in \Sigma$, as follows: $M[\tau_k\rangle M'$ iff $\forall p \in P.\ M'(p) = M(p) \mathbin{<\!\!+} k$.

Let $N = (P, T, F, c, L, \lambda, \Sigma)$ be a LTAPN. We define the corresponding labelled transition system $T(N) = ([P \to \mathcal{B}], L \cup \{\tau_k \mid k \in \Sigma\}, \longrightarrow, \mathcal{E}qv^u)$, where states are markings of N, actions are labels from L together with symbols for time-elapsing, and $M \xrightarrow{a} M'$ iff either $M[t\rangle M'$ and $a = \lambda(t)$, or $M[\tau_k\rangle M'$ and $a = \tau_k$ for some $k \in \Sigma$. For simplicity we define $\mathcal{E}qv^u$ to be the universal relation.

Definition 2. *A* timed-arc BPP *is a LTAPN such that* $|{}^\bullet t| = 1$ *for all* $t \in T$.

Example 4. Consider a timed-arc BPP net $(\{p_1, p_2\}, \{t_1, t_2\}, F, c, \{a, b\}, \lambda, \{1\})$ where F, c and λ are defined in Figure 4. Names of places (circles) are p_1 and p_2 (from left to right) and names of transitions (squares) are t_1 and t_2 (from left to right) such that $\lambda(t_1) = a$ and $\lambda(t_2) = b$. Notice that ${}^\bullet t_1 = \{p_1\}$ and ${}^\bullet t_2 = \{p_2\}$, so the net is indeed a timed-arc BPP. Let $(\{0\}, \emptyset)$ be an initial marking — since $|P| = 2$ we can identify any marking $M : P \to \mathcal{B}$ with

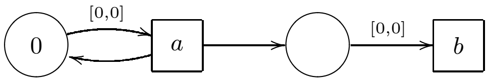

Fig. 4. A timed-arc BPP net

a pair $(M(p_1), M(p_2))$. Now e.g. $(\{0\}, \emptyset) \xrightarrow{a} (\{0\}, \{0\}) \xrightarrow{a} (\{0\}, \{0, 0\}) \xrightarrow{b} (\{0\}, \{0\}) \xrightarrow{\tau_1} (\{1\}, \{1\}) \xrightarrow{\tau_1} (\{2\}, \{2\}) \xrightarrow{\tau_1} \ldots$. Using similar arguments as in Example 2, there cannot be any BPP process bisimilar to the initial marking. On the other hand, for any BPP process there is a timed-arc BPP net bisimilar to it — we use the fact that any BPP process is essentially a Petri net where $|{}^\bullet t| = 1$ for every transition t and then we define all the time constrains as $[0, \infty]$ and set $\Sigma = \emptyset$. So the class of timed-arc BPP is strictly more expressive (w.r.t. bisimilarity) than the BPP class.

Assuming a fixed ordering on $P = \{p_1, \ldots, p_n\}$, there is a natural one-to-one correspondence between $[P \to \mathcal{B}]$ and $\mathcal{B}^n$. Let $M : P \to \mathcal{B}$ then we define $(N_1, \ldots, N_n) \in \mathcal{B}^n$ by $N_i = M(p_i)$ for $1 \leq i \leq n$ and vice versa. In what follows we freely interchange these equivalent notations.

The system $T(N)$ is almost a commutative labelled transition system. There are only two problems: (i) states are not elements from $\mathcal{B}^n_m$ for some fixed $m \in \mathbb{N}_0$ and (ii) the set of actions can be infinite. The following arguments show how to avoid these problems.

Definition 3. *Let $N = (P, T, F, c, L, \lambda, \Sigma)$ be a LTAPN. We define its* maximal guard $mg(N) \in \mathbb{N}_0$ *as the maximal time constraint that appears in N, i.e., $mg(N) = \max\Big(\{t_1, t_2 \mid \exists f \in F|_{P\times T}.\ c(f) = (t_1, t_2)\} \smallsetminus \{\infty\}\Big)$. Let $M \in [P \to \mathcal{B}]$. We define a* compression *of M, $C_M \in [P \to \mathcal{B}_{mg(N)+1}]$, by*

$$C_M(p)(k) = \begin{cases} M(p)(k) & \text{if } k < mg(N) + 1 \\ \sum_{i=mg(N)+1}^{\infty} M(p)(i) & \text{if } k = mg(N) + 1 \\ 0 & \text{if } k > mg(N) + 1. \end{cases}$$

Lemma 6. *Let $N = (P, T, F, c, L, \lambda, \Sigma)$ be a LTAPN and $M_1, M_2 \in [P \to \mathcal{B}]$. If $C_{M_1} = C_{M_2}$ then $M_1 \sim_{T(N)} M_2$.*

Proof. It is a routine exercise to verify that $R = \{(M_1, M_2) \in [P \to \mathcal{B}] \times [P \to \mathcal{B}] \mid C_{M_1} = C_{M_2}\}$ is a bisimulation. $\square$

Let $N = (P, T, F, c, L, \lambda, \Sigma)$ be a LTAPN. By m we denote the number $mg(N) + 1$. We define a commutative transition system $T^c(N) = (\mathcal{B}^n_m, L \cup \{\tau_k \mid k \in \Sigma \ \wedge\ k < m\} \cup T_m, \longrightarrow, \mathcal{E}qv^u)$ where $T_m = \{\tau_m\}$ if there is $k \in \Sigma$ such that $k \geq m$, otherwise $T_m = \emptyset$ (note that the construction of T_m is effective since Σ is a recursive set). We define $M \xrightarrow{a} M'$ for $M, M' \in \mathcal{B}^n_m$ iff either (i) $M[t\rangle M'$ and $a = \lambda(t)$, or (ii) $M[\tau_k\rangle M''$ where $m > k \in \Sigma$ or $\tau_k \in T_m$, such that $M' = C_{M''}$ and $a = \tau_k$.

Proposition 3. *Let N be a LTAPN and M_1, M_2 a pair of markings on N. Then $M_1 \sim_{T(N)} M_2$ iff $C_{M_1} \sim_{T^c(N)} C_{M_2}$.*

Proof. Immediately from Lemma 6. Also note that in $T(N)$ for any $k \geq m = mg(N) + 1$ holds that if $M \xrightarrow{\tau_m} M'$ and $M \xrightarrow{\tau_k} M''$, then $C_{M'} = C_{M''}$. □

Theorem 5. *Given a timed-arc BPP net $N = (P, T, F, c, L, \lambda, \Sigma)$ and a pair of markings M_1, M_2 on N, it is decidable whether $M_1 \sim_{T(N)} M_2$.*

Proof. First show that $T^c(N)$ is an ECTS. Then by Proposition 3 and Theorem 1. For details see [22]. □

Remark 3. It remains an open problem whether bisimilarity is decidable for timed-arc BPP with continuous time, i.e., if we allow e.g. $\Sigma = \mathbb{R}_0^+$.

4 Conclusion

We suggested a subclass of labelled transition systems called effective commutative transition systems (ECTS) where bisimilarity is decidable, and we showed that semantics of many extensions of BPP process algebra can be defined within the ECTS class. This approach seems to be feasible also for other natural extensions of BPP: the crucial condition to be satisfied is probably (6), saying that $\sim_k$ are congruences. This condition fails e.g. for Petri nets, and indeed strong bisimilarity becomes undecidable here [15].

Decidability of weak bisimilarity of BPP is still a well known open problem. Here the problematic condition is (5), stating that the transition system is image-finite, which is not the case for weak bisimilarity. Nevertheless, we can still instead of potentially infinite set of successors $\mathsf{next}(\alpha, a)$ examine only its finite subset such that soundness and completeness of the tableau system is preserved. This possibility was exploited by Stirling in [24] for weak bisimilarity of normed BPP, however, with additional technical restrictions. To design finite subsets of $\mathsf{next}(\alpha, a)$ preserving soundness and completeness even in the general case might be a reasonable way to attack this problem.

Acknowledgement. I would like to thank my advisor Mogens Nielsen for his kind supervision. I also thank Daniel Polansky, Jan Strejcek and the referees for their comments and suggestions.

References

1. P.A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.
2. J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, IX(2):127–168, 1986.
3. J.C.M. Baeten and J.A. Bergstra. Mode transfer in process algebra. Technical report CSR 00-01, Vakgroep Informatica, Technische Universiteit Eindhoven, 2000.

4. B. Berard, A. Labroue, and Ph. Schnoebelen. Verifying performance equivalence for timed basic parallel processes. In *Proc. of FOSSACS'2000*, volume 1784 of *LNCS*, p. 35–47. Springer-Verlag, 2000.
5. J.A. Bergstra. A mode transfer operator in process algebra. Technical report P8808b, University of Amsterdam, The Netherlands, 1989.
6. T. Bolognesi, F. Lucidi, and S. Trigila. From timed Petri nets to timed LOTOS. In *Proc. of the IFIP WG 6.1 10th International Symposium on Protocol Specification, Testing and Verification*, p. 1–14. Amsterdam, 1990.
7. A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. In *Proc. of STACS'99*, volume 1563 of *LNCS*, p. 323–333. Springer-Verlag, 1999.
8. O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification on infinite structures. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, chapter 9, p. 545–623. Elsevier Science, 2001.
9. S. Christensen. *Decidability and Decomposition in Process Algebras.* PhD thesis, The University of Edinburgh, 1993.
10. S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation is decidable for basic parallel processes. In *Proc. of CONCUR'93*, volume 715 of *LNCS*, p. 143–157. Springer-Verlag, 1993.
11. T. Cobben and A. Engels. Disrupt and interrupt in MSC: Possibilities and problems. In *Proc. of the 1st Workshop of the SDL Forum Society on SDL and MSC*, number 104 in Informatik-Berichte, p. 75–83. 1998.
12. L.E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with distinct factors. *American Journal of Mathematics*, 35:413–422, 1913.
13. B. Diertens. New features in PSF I: Interrupts, disrupts, and priorities. Technical report P9417, University of Amsterdam, The Netherlands, 1994.
14. H.M. Hanisch. Analysis of place/transition nets with timed-arcs and its application to batch process control. In *Application and Theory of Petri Nets*, volume 691 of *LNCS*, p. 282–299, 1993.
15. P. Jancar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148(2):281–301, 1995.
16. P. Jancar and F. Moller. Techniques for decidability and undecidability of bisimilarity – an invited tutorial. In *Proc. of CONCUR '99*, volume 1664 of *LNCS*, p. 30–45. Springer-Verlag, 1999.
17. R. Mayr. Process rewrite systems. *Information and Comp.*, 156(1):264–286, 2000.
18. R. Mayr. Undecidable problems in unreliable computations. In *Proc. of LATIN'00*, volume 1776 of *LNCS*. Springer-Verlag, 2000.
19. R. Milner. *Communication and Concurrency.* Prentice-Hall, 1989.
20. V. Valero Ruiz, D. de Frutos Escrig, and O. Marroquin Alonso. Decidability of properties of timed-arc Petri nets. In *ICATPN 2000*, volume 1825 of *LNCS*, p. 187–206. Springer-Verlag, 2000.
21. J. Srba. Basic process algebra with deadlocking states. *Theoretical Computer Science*, 266(1–2):605–630, 2001.
22. J. Srba. Note on the tableau technique for commutative transition systems. Technical Report RS-01-50, BRICS Research Series, 2001.
23. J. Srba. Strong bisimilarity and regularity of basic parallel processes is PSPACE-hard. In *Proc. of STACS'02*, LNCS. Springer-Verlag, 2002. To appear.
24. C. Stirling. Decidability of weak bisimilarity for a subset of basic parallel processes. In *Proc. of FOSSACS'01*, volume 2030 of *LNCS*, p. 379–393. Springer-Verlag, 2001.