ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A study on team bisimulation and H-team bisimulation for BPP nets

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Gorrieri, R. (2022). A study on team bisimulation and H-team bisimulation for BPP nets. THEORETICAL COMPUTER SCIENCE, 897, 83-113 [10.1016/j.tcs.2021.09.037].

Availability:
This version is available at: https://hdl.handle.net/11585/849478 since: 2023-12-15

Published:
DOI: http://doi.org/10.1016/j.tcs.2021.09.037

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

09 March 2025

A Study on Team Bisimulation and H-team Bisimulation for BPP Nets

Roberto Gorrieri*

Dipartimento di Informatica - Scienza e Ingegneria
Università di Bologna,
Mura A. Zamboni 7, 41027 Bologna, Italy

Abstract

A subclass of finite Petri nets, called BPP nets (acronym of *Basic Parallel Processes*), was recently equipped with an efficiently decidable, truly concurrent, bisimulation-based, behavioral equivalence, called *team bisimilarity*. This equivalence is a very intuitive extension of classic bisimulation equivalence (over labeled transition systems) to BPP nets and it is checked in a distributed manner. This paper has three goals. First of all, we provide BPP nets with various causality-based observational semantics, notably a novel semantics, called *causal-net* bisimilarity. Then, we define a variant semantics, called *h-team* bisimilarity, coarser than team bisimilarity, for which we adapt the modal logic characterization and the axiomatization of team bisimilarity. Then, we complete the study about team bisimilarity and h-team bisimilarity, by comparing them with the causality-based semantics we have introduced: the main results are that team bisimilarity coincides with causal-net bisimilarity, while h-team bisimilarity with *fully-concurrent* bisimilarity.

Keywords: Petri nets, BPP process algebra, fully-concurrent bisimulation, team bisimulation, Hennessy-Milner modal logic, axiomatization.

1. Introduction

A BPP net is a simple type of finite Place/Transition Petri net [49, 15, 52, 28] whose transitions have singleton pre-set. BPP is the acronym of *Basic Parallel Processes* [12], a simple CCS [43, 27] fragment (without the restriction operator) whose processes cannot communicate. In [28] a variant of BPP, which requires guarded summation (as in Simple BPP [17], SBPP [20] or BPP_g [12]) and also that the body of each process constant is guarded (i.e., guarded recursion), is actually shown to represent *all and only* the BPP nets, up to net isomorphism, and this explains the name of this class of nets. Hence, we can uniformly compare results achieved on BPP nets or on the BPP subcalculus with guarded summation and guarded recursion. About expressiveness, BPP is

*Corresponding author
Email address: `roberto.gorrieri@unibo.it` (Roberto Gorrieri)

the largest CCS fragment for which bisimilarity [43] is decidable [12]; moreover, used as formal language recognizer, BPP is rather powerful as it can represent all the regular languages, many context-free, and some context-dependent ones [27]. However, as models for distributed systems, BPP nets have limited applicability: even if they can represent distributed systems composed of non-communicating sequential processes (that, nonetheless, can spawn, so that the reachable markings can be infinitely many, as for the *semi-counter* in Example 1), their lack of synchronization (as the transition pre-set is always a singleton) prevents the modeling of many real-life applications.

In a recent paper [30], we proposed a novel behavioral equivalence for BPP nets, based on a suitable generalization of the concept of bisimulation [48, 43], originally defined over labeled transition systems (LTSs, for short) A *team bisimulation* R over the places of an *unmarked* BPP net is a relation such that if two places s_1 and s_2 are related by R, then if (one token in place) s_1 performs a and produces the marking m_1, then (one token in place) s_2 may perform a producing a marking m_2 such that m_1 and m_2 are element-wise, bijectively related by R (and vice versa if s_2 moves first). *Team bisimilarity* is the largest team bisimulation over the places of the *unmarked* BPP net, and then such a relation is lifted to markings by *additive closure*: if place s_1 is team bisimilar to s_2 and the marking m_1 is team bisimilar to m_2 (the base case relates the empty marking to itself), then also $s_1 \oplus m_1$ is team bisimilar to $s_2 \oplus m_2$, where $\_ \oplus \_$ is the operator of multiset union. Note that to check whether two markings are team bisimilar we need not to construct an LTS, such as the *reachability graph* [28], describing the global behavior of the whole system, but only to find a suitable, *bijective*, team bisimilarity-preserving match among the local, sequential states (i.e., the elements of the two markings). In other words, two distributed systems, each composed of a *team* of sequential, non-cooperating processes (i.e., the tokens in the BPP net), are equivalent if it is possible to match each sequential component of the first system with one team-bisimilar, sequential component of the other system, as in any sports where two competing (distributed) teams have the same number of (sequential) players.

A bit surprisingly, the complexity of checking whether two markings are team bisimilar is very low. First, by adapting the classic Kanellakis-Smolka algorithm [37, 38] for standard bisimulation equivalence over LTSs, team bisimulation equivalence over places can be computed in $O(m \cdot p^2 \cdot n)$ time, where m is the number of net transitions, p is the size of the largest post-set (i.e., p is the least natural such that $|t^\bullet| \leq p$ for all t) and n is the number of places. Then, checking whether two markings of size k are team bisimilar can be done in $O(k^2)$ time. Note also that if we need to check whether other two markings of the same net, say m_1' and m_2', are team equivalent, we can reuse the already computed team bisimilarity over places, and so such a verification will take only $O(k^2)$ time, if k is the size of m_1' and m_2'. Of course, we proved that team bisimilar markings respect the global behavior; in particular, the token game (actually, we proved that team bisimilarity implies interleaving bisimilarity) and the causal behavior (actually, we proved that team bisimilarity coincides with *place bisimilarity* [3, 4]).

In this paper we complete the comparison between team bisimilarity on markings and the causal semantics of BPP nets. We propose a novel coinductive semantics, called *causal-net* bisimilarity, inspired by [24], which is essentially a bisimulation semantics over the *causal nets* (also called *occurrence nets*) [25, 6, 47] of the BPP net under scrutiny. We prove that team bisimilarity coincides with causal-net bisimilarity, hence proving that our distributed semantics is coherent with the expected causal semantics of BPP nets. Moreover, we define a slight strengthening of *fully-concurrent* bisimulation [7] (fc-bisimulation, for short), called *state-sensitive* fc-bisimulation, which requires additionally that, for each pair of related processes, the current markings have the same size. We also prove that causal-net bisimilarity coincides with state-sensitive fc-bisimilarity over BPP nets. These behavioral causal semantics have been provided for BPP nets, but they can be easily adapted for general P/T nets [49, 15, 52, 28].

The other main goal of this paper is to show that fc-bisimilarity can be characterized in a team-style, by means of *h-team* bisimulation equivalence (the prefix *h*- is to remind *h*istory-preserving bisimilarity [22], an equivalence that inspired fc-bisimilarity). The essential difference between a team bisimulation and an h-team bisimulation is that the former is a relation on the set of places only, while the latter is a relation on the set composed of the places *and* the empty marking θ. Besides proving that h-team bisimilarity coincides with fc-bisimilarity, in the second part of the paper, we adapt the technical results obtained for team bisimulation equivalence in [30], to h-team bisimulation equivalence. In particular, we show that h-team bisimilarity can be characterized by a simple modal logic, called HTML, which extends conservatively Hennessy-Milner logic (HML) [33, 2]; moreover, we prove that h-team bisimilarity can be axiomatized finitely over the BPP process algebra (with guarded summation and guarded constants).

The paper is organized as follows. Section 2 introduces the basic definitions about BPP nets. Section 3 discusses the causal semantics of BPP nets. First, causal-net bisimulation is introduced, then (strong) fully-concurrent bisimilarity [7], and also the state-sensitive version of this equivalence. Section 4 recalls the main definitions and results about team bisimilarity from [30]; in this section we also prove a novel result: causal-net bisimilarity coincides with team bisimilarity for BPP nets. Section 5 defines h-team bisimulation equivalence and studies its properties; in particular, we first prove that h-team bisimilarity coincides with fc-bisimilarity, then in Section 5.2 we discuss a modal logic characterization of h-team bisimilarity. In Section 6 we describe also a finite, sound and complete, axiomatization of h-team bisimilarity over the process algebra BPP. Finally, Section 7 discusses related literature and some future research.

2. Basic Definitions

Definition 1. (Multiset) Let $\mathbb{N}$ be the set of natural numbers. Given a finite set S, a *multiset* over S is a function $m : S \to \mathbb{N}$. The *support* set $dom(m)$ of m is $\{s \in S \mid m(s) \neq 0\}$. The set of all multisets over S, denoted by $\mathscr{M}(S)$, is ranged over by m. We write $s \in m$ if $m(s) > 0$. The *multiplicity* of s in m is given by the number $m(s)$. The *size* of m, denoted by $|m|$, is the number $\sum_{s \in S} m(s)$, i.e., the total number of its elements. A multiset m such that $dom(m) = \emptyset$ is called *empty* and is denoted by θ. We write $m \subseteq m'$ if $m(s) \leq m'(s)$ for all $s \in S$. *Multiset union* $\_ \oplus \_$ is defined as follows: $(m \oplus m')(s) = m(s) + m'(s)$. *Multiset difference* $\_ \ominus \_$ is defined as follows: $(m_1 \ominus m_2)(s) = max\{m_1(s) - m_2(s), 0\}$. The *scalar product* of a number j with m is the multiset $j \cdot m$ defined as $(j \cdot m)(s) = j \cdot (m(s))$. By s_i we also denote the multiset with s_i as its only element. Hence, a multiset m over $S = \{s_1, \ldots, s_n\}$ can be represented as $k_1 \cdot s_1 \oplus k_2 \cdot s_2 \oplus \ldots \oplus k_n \cdot s_n$, where $k_j = m(s_j) \geq 0$ for $j = 1, \ldots, n$. □

Definition 2. (BPP net) A labeled *BPP net* is a tuple $N = (S, A, T)$, where

- S is the finite set of *places*, ranged over by s (possibly indexed),
- A is the finite set of *labels*, ranged over by ℓ (possibly indexed), and
- $T \subseteq S \times A \times \mathscr{M}(S)$ is the finite set of *transitions*, ranged over by t (possibly indexed).

Given a transition $t = (s, \ell, m)$, we use the notation:

- ${}^\bullet t$ to denote its *pre-set* s (which is a single place) of tokens to be consumed;
- $l(t)$ for its *label* ℓ, and
- $t^\bullet$ to denote its *post-set* m (which is a multiset, possibly even empty) of tokens to be produced.

Hence, transition t can be also represented as ${}^\bullet t \xrightarrow{l(t)} t^\bullet$. We also define pre-sets and post-sets for places as follows: ${}^\bullet s = \{t \in T \mid s \in t^\bullet\}$ and $s^\bullet = \{t \in T \mid s \in {}^\bullet t\}$. Note that while the pre-set (post-set) of a transition is, in general, a multiset, the pre-set (post-set) of a place is a set. $\square$

Graphically, a place is represented by a little circle and a transition by a little box, and these may be connected by directed arcs. These arcs may be labeled with the number representing how many tokens of that type are to be removed from (or produced into) that place; no label on the arc is interpreted as the number one. This numerical label of the arc is called its *weight*.

Definition 3. (Marking, BPP net system) A multiset over S is called a *marking*. Given a marking m and a place s, we say that the place s contains $m(s)$ *tokens*, graphically represented by $m(s)$ bullets inside place s. A *BPP net system* $N(m_0)$ is a tuple (S, A, T, m_0), where (S, A, T) is a BPP net and m_0 is a marking over S, called the *initial marking*. We also say that $N(m_0)$ is a *marked* net. $\square$

Definition 4. (Enabling, firing sequence, reachable place, dynamically reduced) A transition t is *enabled* at marking m, denoted by $m[t\rangle$, if ${}^\bullet t \subseteq m$. The execution (or *firing*) of t enabled at m produces the marking $m' = (m \ominus {}^\bullet t) \oplus t^\bullet$. This is written $m[t\rangle m'$.

A *firing sequence* starting at m is defined inductively as follows:

- $m[\varepsilon\rangle m$ is a firing sequence (where ε denotes an empty sequence of transitions);
- if $m[\sigma\rangle m'$ is a firing sequence and $m'[t\rangle m''$, then $m[\sigma t\rangle m''$ is a firing sequence.

The set of *reachable markings* from m is $[m\rangle = \{m' \mid \exists \sigma. m[\sigma\rangle m'\}$. A BPP net system $N(m_0) = (S, A, T, m_0)$ is *safe* if each marking m reachable from the initial marking m_0 is a set, i.e., $\forall m \in [m_0\rangle, m(s) \leq 1$ for all $s \in S$. The set of *reachable places* from s is $reach(s) = \bigcup_{m \in [s\rangle} dom(m)$. Note that $reach(s)$ is always a finite set, even if $[s\rangle$ is infinite. A BPP net system $N(m_0) = (S, A, T, m_0)$ is *dynamically reduced* if $\forall s \in S$ there exists $m \in [m_0\rangle$. $m(s) \geq 1$ and $\forall t \in T\, \exists m, m' \in [m_0\rangle$ such that $m[t\rangle m'$. $\square$

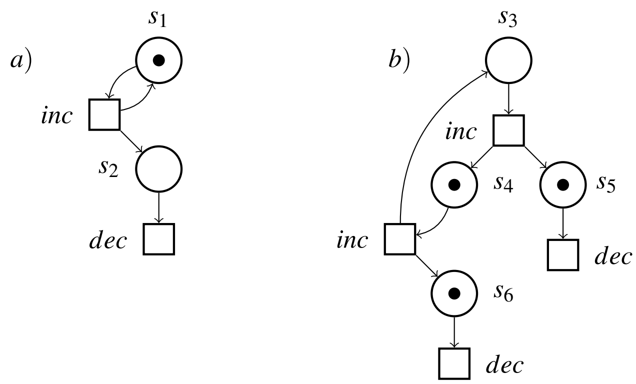

Figure 1: The net representing a semi-counter in (a), and a variant in (b)

Example 1. By using the drawing convention mentioned above, Figure 1 shows in (a) the simplest BPP net representing a semi-counter, i.e., a counter which cannot test for zero. Note that the number represented by this semi-counter is given by the number of tokens which are present in place s_2, i.e., in the place ready to perform *dec*; hence, Figure 1(a) represents a semi-counter holding number 0; note also that the number of tokens which can be accumulated in s_2 is unbounded. Indeed, the set of reachable markings for a BPP net can be countably infinite. In (b), a variant semi-counter is outlined, which holds number 2 (i.e., two tokens are ready to perform *dec*). □

Now we recall a popular behavioral equivalence relation: interleaving bisimilarity.

Definition 5. (Interleaving Bisimulation) Let $N = (S, A, T)$ be a BPP net. An *interleaving bisimulation* is a relation $R \subseteq \mathcal{M}(S) \times \mathcal{M}(S)$ such that if $(m_1, m_2) \in R$ then

- $\forall t_1$ such that $m_1[t_1\rangle m_1'$, $\exists t_2$ such that $m_2[t_2\rangle m_2'$ with $l(t_1) = l(t_2)$ and $(m_1', m_2') \in R$,
- $\forall t_2$ such that $m_2[t_2\rangle m_2'$, $\exists t_1$ such that $m_1[t_1\rangle m_1'$ with $l(t_1) = l(t_2)$ and $(m_1', m_2') \in R$.

Two markings m_1 and m_2 are *interleaving bisimilar*, denoted by $m_1 \sim_{int} m_2$, if there exists an interleaving bisimulation R such that $(m_1, m_2) \in R$. □

Interleaving bisimilarity $\sim_{int}$, which is defined as the union of all the interleaving bisimulations, is the largest interleaving bisimulation and also an equivalence relation.

Example 2. Continuing Example 1 about Figure 1, it is easy to realize that relation $R = \{(s_1 \oplus k \cdot s_2, s_3 \oplus k_1 \cdot s_5 \oplus k_2 \cdot s_6) \mid k = k_1 + k_2 \text{ and } k, k_1, k_2 \geq 0\} \cup \{(s_1 \oplus k \cdot s_2, s_4 \oplus k_1 \cdot s_5 \oplus k_2 \cdot s_6) \mid k = k_1 + k_2 \text{ and } k, k_1, k_2 \geq 0\}$ is an interleaving bisimulation. Since $(s_1, s_3) \in R$, we have that $s_1 \sim_{int} s_3$. □

Remark 1. (Complexity of $\sim_{int}$) The problem of checking whether two markings of a BPP net are interleaving bisimilar is roughly exponential in time; more precisely, this decision problem is PSPACE-complete [35] (w.r.t. the size of the net, where the *size of a BPP net* is the sum of the number of its places and of its transitions). □

3. Causality-Based Semantics

We start with the most concrete equivalence definable over BPP nets: isomorphism.

Definition 6. (Isomorphism) Given two BPP nets $N_1 = (S_1, A, T_1)$ and $N_2 = (S_2, A, T_2)$, we say that N_1 and N_2 are *isomorphic via f* if there exists a type-preserving bijection $f : S_1 \cup T_1 \to S_2 \cup T_2$ (i.e., a bijection such that $f(S_1) = S_2$ and $f(T_1) = T_2$), satisfying the following condition:

$\forall t \in T_1$, if $t = ({}^\bullet t, \ell, t^\bullet)$, then $f(t) = (f({}^\bullet t), \ell, f(t^\bullet))$,

where f is homomorphically extended to markings: $f(\theta) = \theta$ and $f(m_1 \oplus m_2) = f(m_1) \oplus f(m_2)$ (i.e., f is applied element-wise to each element of the marking).

Two BPP net systems $N_1(m_1)$ and $N_2(m_2)$ are *rooted isomorphic* if the isomorphism f ensures, additionally, that $f(m_1) = m_2$. □

In order to define our approach to causality-based semantics for BPP nets, we need some auxiliary definitions, adapting those in, e.g., [46, 25, 6, 7, 47, 24].

Definition 7. (Acyclic net) A BPP net $N = (S, A, T)$ is *acyclic* if there exists no sequence $x_1 x_2 \ldots x_n$ such that $n \geq 3$, $x_i \in S \cup T$ for $i = 1, \ldots, n$, $x_1 = x_n$, $x_1 \in S$ and $x_i \in {}^\bullet x_{i+1}$ for $i = 1, \ldots, n-1$, i.e., the arcs of the net do not form any cycle. □

The concurrent semantics of a marked P/T net is defined by a class of particular acyclic safe nets, where places are not branched (hence they are essentially deterministic) and all arcs have weight 1. This kind of net is called *causal net*. We use the name C (possibly indexed) to denote a causal net, the set B to denote its places (called *conditions*), the set E to denote its transitions (called *events*), and L to denote its labels.

Definition 8. (Causal net) A causal net is a marked BPP net $C(\mathsf{m}_0) = (B, L, E, \mathsf{m}_0)$ satisfying the following conditions:

1. C is acyclic;
2. $\forall b \in B \;\; |{}^\bullet b| \leq 1 \wedge |b^\bullet| \leq 1$ (i.e., the places are not branched);
3. $\forall b \in B \;\; \mathsf{m}_0(b) = \begin{cases} 1 & \text{if } {}^\bullet b = \emptyset \\ 0 & \text{otherwise;} \end{cases}$
4. $\forall e \in E \;\; e^\bullet(b) \leq 1$ for all $b \in B$ (i.e., all the arcs have weight 1).

We denote by $Min(C)$ the set m_0, and by $Max(C)$ the set $\{b \in B \mid b^\bullet = \emptyset\}$. □

Note that a BPP causal net, being a BPP net, is finite; since it is acyclic, it represents a finite computation. Note also that any reachable marking of a BPP causal net is a set, i.e., this net is *safe*; in fact, the initial marking is a set and, assuming by induction that a reachable marking m is a set and enables t, i.e., $\mathsf{m}[t\rangle \mathsf{m}'$, then also $\mathsf{m}' = (\mathsf{m} \ominus {}^\bullet t) \oplus t^\bullet$ is a set, because the net is acyclic and because of the condition on the shape of the post-set of t (weights can only be 1).

As the initial marking of a causal net is fixed by its shape (according to item 3 of Definition 8), in the following, in order to make the notation lighter, we often omit the indication of the initial marking, so that $C(\mathsf{m}_0)$ is simply denoted by C.

Definition 9. (Moves of a causal net) Given two BPP causal nets $C = (B, L, E, \mathsf{m}_0)$ and $C' = (B', L, E', \mathsf{m}_0)$, we say that C moves in one step to C' through e, denoted by $C[e\rangle C'$, if ${}^\bullet e \subseteq Max(C)$, $E' = E \cup \{e\}$ and $B' = B \cup e^\bullet$. □

Definition 10. (Folding and Process) A *folding* from a BPP causal net $C = (B, L, E, \mathsf{m}_0)$ into a BPP net system $N(m_0) = (S, A, T, m_0)$ is a function $\rho : B \cup E \to S \cup T$, which is type-preserving, i.e., such that $\rho(B) \subseteq S$ and $\rho(E) \subseteq T$, satisfying the following:

- $L = A$ and $\mathsf{l}(e) = l(\rho(e))$ for all $e \in E$;
- $\rho(\mathsf{m}_0) = m_0$, i.e., $m_0(s) = |\rho^{-1}(s) \cap \mathsf{m}_0|$;
- $\forall e \in E, \rho({}^\bullet e) = {}^\bullet \rho(e)$, i.e., $\rho({}^\bullet e)(s) = |\rho^{-1}(s) \cap {}^\bullet e|$ for all $s \in S$;
- $\forall e \in E$, $\rho(e^\bullet) = \rho(e)^\bullet$, i.e., $\rho(e^\bullet)(s) = |\rho^{-1}(s) \cap e^\bullet|$ for all $s \in S$.

A pair (C, ρ), where C is a BPP causal net and ρ a folding from C to a BPP net system $N(m_0)$, is a *process* of $N(m_0)$, written also as π. □

Definition 11. (Isomorphic processes) Given a BPP net system $N(m_0)$, two of its processes (C_1, ρ_1) and (C_2, ρ_2) are *isomorphic via f* if C_1 and C_2 are rooted isomorphic via bijection f (see Definition 6) and $\rho_1 = \rho_2 \circ f$. □

Definition 12. (Moves of a process) Let $N(m_0) = (S, A, T, m_0)$ be a net system and let (C_i, ρ_i), for $i = 1, 2$, be two processes of $N(m_0)$. We say that (C_1, ρ_1) moves in one step to (C_2, ρ_2) through e, denoted by $(C_1, \rho_1) \xrightarrow{e} (C_2, \rho_2)$, if $C_1[e\rangle C_2$ and $\rho_1 \subseteq \rho_2$.
If $\pi_1 = (C_1, \rho_1)$ and $\pi_2 = (C_2, \rho_2)$ and $(C_1, \rho_1) \stackrel{e}{\longmapsto} (C_2, \rho_2)$, we also denote the move as $\pi_1 \xrightarrow{t} \pi_2$, where $\rho_2(e) = t$. □

Definition 13. (Partial orders of events from a process) From a causal net $C = (B, L, E)$, we can extract the *partial order of its events* $\mathsf{E_C} = (E, \preceq)$, where $e_1 \preceq e_2$ iff there exists a sequence $x_1 x_2 x_3 \dots x_n$ such that $n \geq 3$, $x_i \in B \cup E$ for $i = 1, \dots, n$, $e_1 = x_1, e_2 = x_n$, and $x_i \in {}^\bullet x_{i+1}$ for $i = 1, \dots, n-1$; in other words, $e_1 \preceq e_2$ if there is a path from e_1 to e_2. Two partial orders $(E_1, \preceq_1)$ and $(E_2, \preceq_2)$ are isomorphic if there is a label-preserving, order-preserving bijection $g : E_1 \to E_2$, i.e., a bijection such that $\mathsf{l}_1(e) = \mathsf{l}_2(g(e))$ and $e \preceq_1 e'$ if and only if $g(e) \preceq_2 g(e')$. We also say that g is an *event isomorphism* between the causal nets C_1 and C_2 if it is an isomorphism between their associated partial orders of events E_{C_1} and E_{C_2}. □

We now define two well-known linear-time causality-based equivalences for BPP nets. The first one, known in the literature as *causal equivalence* [47], is named *causal-trace equivalence* and equates two markings of a net from which the same causal nets originate. The latter is more abstract as it forgets about the places of the causal nets and keeps only the partial order of events, yielding *partial-order-trace equivalence*.

Definition 14. (Causal-trace equivalence) Let $N = (S, A, T)$ be a BPP net. For a marking m, let $Cn[m] = \{C \mid \exists \rho.(C, \rho)$ is a process of $N(m)\}$ be the set of causal nets for the BPP net system $N(m)$. Two markings m_1 and m_2 are *causal-trace equivalent*, denoted by $m_1 =_{ct} m_2$, if $Cn[m_1] = Cn[m_2]$. □

Definition 15. (Partial-order-trace equivalence) Let $N = (S, A, T)$ be a BPP net. For a marking m, let $P[m] = \{\mathsf{E}_C \mid \exists \rho.(C, \rho) \text{ is a process of } N(m)\}$ be the set of partial orders for the BPP net system $N(m)$. Two markings m_1 and m_2 are *partial-order-trace equivalent*, denoted by $m_1 =_{pt} m_2$, if $P[m_1] \equiv P[m_2]$, i.e., if $\forall \mathsf{E}_{C_1} \in P[m_1] \exists \mathsf{E}_{C_2} \in P[m_2]$ such that E_{C_1} and E_{C_2} are isomorphic, and vice versa, $\forall \mathsf{E}_{C_2} \in P[m_2] \exists \mathsf{E}_{C_1} \in P[m_1]$ such that E_{C_1} and E_{C_2} are isomorphic. □

Proposition 1. (Causal-trace equivalence is finer than partial-order-trace equivalence) Given a BPP net $N = (S, A, T)$, for each $m_1, m_2 \in \mathscr{M}(S)$, if $m_1 =_{ct} m_2$ then $m_1 =_{pt} m_2$.

PROOF. If $m_1 =_{ct} m_2$, then $Cn[m_1] = Cn[m_2]$. So, $P[m_1] = P[m_2]$, i.e., $m_1 =_{pt} m_2$. □

3.1. Causal-Net Bisimulation

We want to define a bisimulation-based equivalence which is coarser than the very concrete, branching-time semantics of *occurrence net equivalence* [21] (where two nets are occurrence net equivalent if and only if they have isomorphic *unfoldings* [45, 16]), and finer than the linear-time semantics of causal-trace equivalence. The proposed novel behavioral equivalence is the following *causal-net bisimulation*, inspired by [24].

Definition 16. (Causal-net bisimulation) Let $N = (S, A, T)$ be a BPP net. A *causal-net bisimulation* is a relation R, composed of triples of the form (ρ_1, C, ρ_2), where, for $i = 1, 2$, (C, ρ_i) is a process of $N(m_{0_i})$ for some m_{0_i}, such that if $(\rho_1, C, \rho_2) \in R$ then

i) $\forall t_1, C', \rho_1'$ such that $(C, \rho_1) \xrightarrow{e} (C', \rho_1')$ with $\rho_1'(e) = t_1$, there exist t_2, ρ_2' such that $(C, \rho_2) \xrightarrow{e} (C', \rho_2')$, with $\rho_2'(e) = t_2$, and $(\rho_1', C', \rho_2') \in R$;

ii) and symmetrically, $\forall t_2, C', \rho_2'$ such that $(C, \rho_2) \xrightarrow{e} (C', \rho_2')$ with $\rho_2'(e) = t_2$, there exist t_1, ρ_1' such that $(C, \rho_1) \xrightarrow{e} (C', \rho_1')$, with $\rho_1'(e) = t_1$, and $(\rho_1', C', \rho_2') \in R$.

Two markings m_1 and m_2 of N are cn-bisimilar (or cn-bisimulation equivalent), denoted by $m_1 \sim_{cn} m_2$, if there exists a causal-net bisimulation R containing a triple $(\rho_1^0, C^0, \rho_2^0)$, where C^0 contains no events and $\rho_i^0(Min(C^0)) = \rho_i^0(Max(C^0)) = m_i$ for $i = 1, 2$. □

Let us denote by $\sim_R^{cn} = \{(m_1, m_2) \mid m_1 \text{ is cn-bisimilar to } m_2 \text{ thanks to } R\}$. Of course, cn-bisimilarity $\sim_{cn}$ can be seen as $\bigcup \{\sim_R^{cn} \mid R \text{ is a causal-net bisimulation}\} = \sim_{\mathscr{R}}^{cn}$, where $\mathscr{R} = \bigcup \{R \mid R \text{ is a causal-net bisimulation}\}$ is the largest causal-net bisimulation by item 3 of the following obvious proposition.

Proposition 2. For each BPP net $N = (S, A, T)$, the following hold:

1. the identity relation $\mathscr{I} = \{(\rho, C, \rho) \mid \exists m \in \mathscr{M}(S).(C, \rho) \text{ is a process of } N(m)\}$ is a causal-net bisimulation;
2. the inverse relation $R^{-1} = \{(\rho_2, C, \rho_1) \mid (\rho_1, C, \rho_2) \in R\}$ of a causal-net bisimulation R is a causal-net bisimulation;
3. the union $\bigcup_{i \in I} R_i$ of causal-net bisimulations R_i is a causal-net bisimulation. □

More problematic is to define the relational composition $R_1 \circ R_2$ of two causal-net bisimulations R_1 and R_2. One trivial possibility is to state that

$R_1 \circ R_2 = \{(\rho_1, C, \rho_3) \mid \exists \rho_2.\ (\rho_1, C, \rho_2) \in R_1 \wedge (\rho_2, C, \rho_3) \in R_2\}$

but this definition would prevent the composition of cn-bisimulations that differ only for the actual choice of the causal net C; in other words, R_2 may be using a causal net $\overline{C}$ which is simply isomorphic to C. Hence, the more generous definition of composition we are proposing is the following. Given a BPP net $N = (S, A, T)$, the composition, up to net isomorphism, of two causal-net bisimulations R_1 and R_2 is the relation

$R_1 \circ R_2 = \{(\rho_1, C, \rho_3) \mid \exists \rho_2.\ (\rho_1, C, \rho_2) \in R_1 \wedge \exists (\overline{\rho}_2, \overline{C}, \overline{\rho}_3) \in R_2$ such that (C, ρ_2) and $(\overline{C}, \overline{\rho}_2)$ are isomorphic processes via $f \wedge \rho_3 = \overline{\rho}_3 \circ f\}$.

Note that this definition is intuitively correct because we are requiring that (C, ρ_2) and $(\overline{C}, \overline{\rho}_2)$ are isomorphic processes via f (cf. Definition 11), so that they are related to the same initial marking. The following proposition shows that $R_1 \circ R_2$ is a causal-net bisimulation, indeed.

Proposition 3. For each BPP net $N = (S, A, T)$, the relational composition, up to net isomorphism, $R_1 \circ R_2 = \{(\rho_1, C, \rho_3) \mid \exists \rho_2.\ (\rho_1, C, \rho_2) \in R_1 \wedge \exists (\overline{\rho}_2, \overline{C}, \overline{\rho}_3) \in R_2$ such that (C, ρ_2) and $(\overline{C}, \overline{\rho}_2)$ are isomorphic processes via $f \wedge \rho_3 = \overline{\rho}_3 \circ f\}$ of two causal-net bisimulations R_1 and R_2 is a causal-net bisimulation.

PROOF. Assume that $(\rho_1, C, \rho_3) \in R_1 \circ R_2$ and that $(C, \rho_1) \xrightarrow{e} (C', \rho_1')$ with $\rho_1'(e) = t_1$. Since R_1 is a causal-net bisimulation and $(\rho_1, C, \rho_2) \in R_1$, we have that there exist t_2, ρ_2' such that $(C, \rho_2) \xrightarrow{e} (C', \rho_2')$, with $\rho_2'(e) = t_2$, and $(\rho_1', C', \rho_2') \in R_1$. Since (C, ρ_2) and $(\overline{C}, \overline{\rho}_2)$ are isomorphic via f, it follows that $(\overline{C}, \overline{\rho}_2) \xrightarrow{e'} (\overline{C}', \overline{\rho}_2')$, with $\overline{\rho}_2'(e') = t_2$, where (C', ρ_2') and $(\overline{C}', \overline{\rho}_2')$ are isomorphic via f', where f' extends f in the obvious way (i.e., by mapping event e to e').

As $(\overline{\rho}_2, \overline{C}, \overline{\rho}_3) \in R_2$ and R_2 is a causal-net bisimulation, for $(\overline{C}, \overline{\rho}_2) \xrightarrow{e'} (\overline{C}', \overline{\rho}_2')$, with $\overline{\rho}_2'(e') = t_2$, there exist $t_3, \overline{\rho}_3'$ such that $(\overline{C}, \overline{\rho}_3) \xrightarrow{e'} (\overline{C}', \overline{\rho}_3')$, with $\overline{\rho}_3'(e') = t_3$, and $(\overline{\rho}_2', \overline{C}', \overline{\rho}_3') \in R_2$.

As $\rho_3 = \overline{\rho}_3 \circ f$, it follows that (C, ρ_3) and $(\overline{C}, \overline{\rho}_3)$ are isomorphic via f. Therefore, $(C, \rho_3) \xrightarrow{e} (C', \rho_3')$ is derivable, too, where $\rho_3'(e) = t_3$ and $\rho_3' = \overline{\rho}_3' \circ f'$, so that (C', ρ_3') and $(\overline{C}', \overline{\rho}_3')$ are isomorphic via f'.

Summing up, if $(\rho_1, C, \rho_3) \in R_1 \circ R_2$ and $(C, \rho_1) \xrightarrow{e} (C', \rho_1')$, with $\rho_1'(e) = t_1$, then $\exists t_3, \rho_3'$ such that $(C, \rho_3) \xrightarrow{e} (C', \rho_3')$, with $\rho_3'(e) = t_3$, and $(\rho_1', C', \rho_3') \in R_1 \circ R_2$

The symmetric case when (C, ρ_3) moves first is analogous, hence omitted. Therefore, $R_1 \circ R_2$ is a causal-net bisimulation, indeed. □

Proposition 4. For each BPP net $N = (S, A, T)$, relation $\sim_{cn} \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$ is an equivalence relation.

PROOF. Reflexivity is easy: the identity relation $\mathscr{I} = \{(\rho, C, \rho) \mid \exists m \in \mathscr{M}(S).(C, \rho)$ is a process of $N(m)\}$ is a causal-net bisimulation by Proposition 2(1). Hence, $m \sim_{cn} m$ for all m.

Symmetry derives from the following argument. For any $(m_1, m_2) \in \sim_{cn}$, there exists a causal-net bisimulation R containing a triple $(\rho_1^0, C^0, \rho_2^0)$, where C^0 contains no

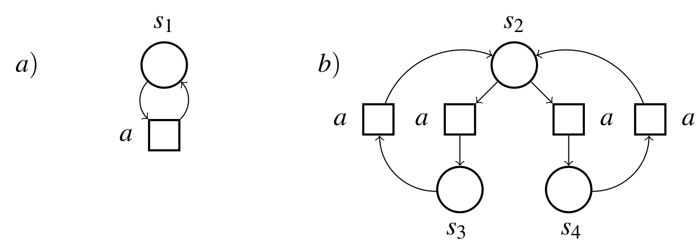

Figure 2: Two cn-bisimilar BPP nets

transitions and $\rho_i^0(Min(C)) = m_i$ for $i = 1,2$. By Proposition 2(2), relation R^{-1} is a causal-net bisimulation containing the triple $(\rho_2^0, C^0, \rho_1^0)$, and so $(m_2, m_1) \in \sim_{cn}$.

Transitivity also holds for $\sim_{cn}$. Assume $(m_1, m_2) \in \sim_{cn}$ and $(m_2, m_3) \in \sim_{cn}$; hence, there exist two causal-net bisimulations R_1 and R_2 such that R_1 has a triple $(\rho_1^0, C^0, \rho_2^0)$, where C^0 has no transitions and $\rho_i^0(Min(C^0)) = m_i$ for $i = 1,2$, and R_2 contains a triple $(\overline{\rho}_2^0, \overline{C}^0, \overline{\rho}_3^0)$, where $\overline{C}^0$ has no transition and $\overline{\rho}_i^0(Min(\overline{C}^0)) = m_i$ for $i = 2,3$. Therefore, (C^0, ρ_2^0) and $(\overline{C}^0, \overline{\rho}_2^0)$ are isomorphic via some bijection f and $\rho_3^0 = \overline{\rho}_3^0 \circ f$. By Proposition 3, relation $R_1 \circ R_2$ is a causal-net bisimulation containing the triple $(\rho_1^0, C^0, \rho_3^0)$; hence, $(m_1, m_3) \in \sim_{cn}$. □

Example 3. Consider the nets in Figure 1. Clearly the net in a) with initial marking s_1 and the net in b) with initial marking s_3 are not isomorphic; however, it is possible to prove that they have isomorphic unfoldings [45, 21, 16]; moreover, it is clear that $s_1 \sim_{cn} s_3$, even if a causal-net bisimulation proving this is not easy to define and would contain infinitely many triples. □

Example 4. Consider the nets in Figure 2. Of course, the initial markings s_1 and s_2 do not generate isomorphic unfoldings; however, $s_1 \sim_{cn} s_2$, even if a causal-net bisimulation proving this would contain infinitely many triples. □

Proposition 5. (Causal-net bisimilarity is finer than causal-trace equivalence) Given a BPP net $N = (S, A, T)$, for each $m_1, m_2 \in \mathscr{M}(S)$, if $m_1 \sim_{cn} m_2$ then $m_1 =_{ct} m_2$.

PROOF. If $m_1 \sim_{cn} m_2$, then for each $C \in Cn[m_1]$, there must be a triple $(\rho_1, C, \rho_2) \in \sim_{cn}$ such that $\rho_1(Min(C)) = m_1$ and $\rho_2(Min(C)) = m_2$. This means that $C \in Cn[m_2]$ and so $Cn[m_1] \subseteq Cn[m_2]$. By a symmetric argument, we can prove $Cn[m_2] \subseteq Cn[m_1]$, so that we conclude that $Cn[m_1] = Cn[m_2]$, i.e., $m_1 =_{ct} m_2$. □

The implication above is strict, as illustrated in the following example.

Example 5. Let us consider the nets in Figure 3. Of course, $s_1 \not\sim_{cn} s_3$, even if they generate the same causal nets, i.e. $s_1 =_{ct} s_3$. In fact, transition $s_1 \xrightarrow{a} s_2$ might be matched by s_3 either with $s_3 \xrightarrow{a} s_4$ or with $s_3 \xrightarrow{a} s_5$, so that it is necessary that $s_2 \sim_{cn} s_4$ or $s_2 \sim_{cn} s_5$; but this is impossible, because only s_2 can perform both b and c. Moreover, $s_6 \not\sim_{cn} s_8$ because they do not generate the same causal nets, i.e., $s_6 \neq_{ct} s_8$. □

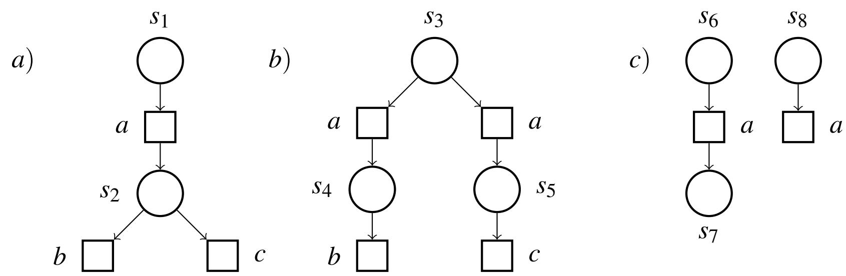

Figure 3: Some non-cn-bisimilar BPP nets

3.2. (State-Sensitive) Fully-Concurrent Bisimulation

In the theory of equivalences for distributed systems, only the events performed are usually considered as the relevant part of the behavior of a system. Hence, causal-net bisimulation, which also observes the structure of the distributed state, may be considered too concrete as an equivalence. We disagree with this view, as the structure of the distributed state is not less observable than the events this distributed state can perform. However, some equivalences have been proposed with this weaker assumption, the most prominent being *fully-concurrent bisimulation* (fc-bisimulation, for short) [7], whose definition was inspired by previous equivalences on other models of concurrency: *history-preserving bisimulation*, originally defined in [51] under the name of *behavior-structure bisimulation*, and then elaborated on in [22] (who called it by this name) and [14] (who called it by *mixed ordering bisimulation*). Besides fc-bisimulation equivalence, we define also a novel, slightly stronger version, called *state-sensitive* fc-bisimulation equivalence, that we prove to coincide with causal-net bisimilarity.

Definition 17. (Fully-concurrent bisimulation) Let $N = (S, A, T)$ be a BPP net. A (strong) *fc-bisimulation* is a relation R, composed of triples of the form (π_1, g, π_2), where, for $i = 1,2$, $\pi_i = (C_i, \rho_i)$ is a process of $N(m_{0i})$ for some m_{0i} and g is an event isomorphism between E_{C_1} and E_{C_2}, such that if $(\pi_1, g, \pi_2) \in R$ then

i) $\forall t_1, \pi_1'$ such that $\pi_1 \xrightarrow{e_1} \pi_1'$ with $\rho_1'(e_1) = t_1$, $\exists t_2, \pi_2', g'$ such that

1. $\pi_2 \xrightarrow{e_2} \pi_2'$ with $\rho_2'(e_2) = t_2$;
2. $g' = g \cup \{(e_1, e_2)\}$, and finally,
3. $(\pi_1', g', \pi_2') \in R$;

ii) and symmetrically, if π_2 moves first.

Two markings m_1 and m_2 are fc-bisimilar, denoted by $m_1 \sim_{fc} m_2$, if there exists an fc-bisimulation R containing a triple $((C_1^0, \rho_1^0), g_0, (C_2^0, \rho_2^0))$, where C_i^0 contains no transitions, g_0 is empty and $\rho_i^0(Min(C_i^0)) = \rho_i^0(Max(C_i^0)) = m_i$ for $i = 1,2$. □

Let us denote by $\sim_R^{fc} = \{(m_1, m_2) \mid m_1$ is fc-bisimilar to m_2 thanks to $R\}$. Of course, $\sim_{fc} = \bigcup\{\sim_R^{fc} \mid R$ is a fully-concurrent bisimulation$\} = \sim_{\mathscr{R}}^{fc}$, where relation

$\mathscr{R} = \bigcup\{R \mid R$ is a fully-concurrent bisimulation$\}$

is the largest fully-concurrent bisimulation by item 4 of the following proposition.

Proposition 6. For each BPP net $N = (S,A,T)$, the following hold:

1. the identity relation $\mathscr{I} = \{((C,\rho),id,(C,\rho)) \mid \exists m.(C,\rho)$ is a process of $N(m)$ and id is the identity event isomorphism on $C\}$ is an fc-bisimulation;
2. the inverse relation $R^{-1} = \{((C_2,\rho_2),g^{-1},(C_1,\rho_1)) \mid ((C_1,\rho_1),g,(C_2,\rho_2)) \in R\}$ of an fc-bisimulation R is an fc-bisimulation;
3. the relational composition, up to isomorphism, $R_1 \circ R_2 = \{((C_1,\rho_1),g,(\overline{C}_3,\overline{\rho}_3)) \mid ((C_1,\rho_1),g_1,(C_2,\rho_2)) \in R_1 \wedge ((\overline{C}_2,\overline{\rho}_2),g_2,(\overline{C}_3,\overline{\rho}_3)) \in R_2 \wedge (C_2,\rho_2)$ and $(\overline{C}_2,\overline{\rho}_2)$ are isomorphic processes via $f_2 \wedge g = g_2 \circ (f_2 \circ g_1)\}$ of two fc-bisimulations R_1 and R_2 is an fc-bisimulation;
4. the union $\bigcup_{i \in I} R_i$ of a family of fc-bisimulations R_i is an fc-bisimulation.

PROOF. The proof of cases 1, 2, and 4 is trivial. The proof for case 3 is similar to the proof of Proposition 3, and so omitted. □

Proposition 7. For each BPP net $N = (S,A,T)$, relation $\sim_{fc} \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$ is an equivalence relation.

PROOF. Similar to the proof of Proposition 4 and so omitted.

Proposition 8. (Fc-bisimilarity implies partial-order-trace equivalence) Given a BPP net $N = (S,A,T)$, for each $m_1, m_2 \in \mathscr{M}(S)$, if $m_1 \sim_{fc} m_2$ then $m_1 =_{pt} m_2$.

PROOF. If $m_1 \sim_{fc} m_2$, then for each $\mathsf{E}_{C_1} \in P[m_1]$, there must be a triple $((C_1,\rho_1),g,(C_2,\rho_2)) \in \sim_{fc}$ such that, for $i = 1,2$, $\rho_i(Min(C_i)) = m_i$, E_{C_i} is the partial order of C_i and g is an isomorphism between E_{C_1} and E_{C_2}. So $\mathsf{E}_{C_2} \in P[m_2]$. Hence, $\forall \mathsf{E}_{C_1} \in P[m_1]$, $\exists \mathsf{E}_{C_2} \in P[m_2]$ such that the two partial orders are isomorphic. By a symmetric argument, we can prove that $\forall \mathsf{E}_{C_2} \in P[m_2]$, $\exists \mathsf{E}_{C_1} \in P[m_1]$ such that the two partial orders are isomorphic. Hence, we conclude that $P[m_1] \equiv P[m_2]$, i.e., $m_1 =_{pt} m_2$. □

Proposition 9. (Cn-bisimilarity is finer than fc-bisimilarity) For each BPP net $N = (S,A,T)$, for each $m_1, m_2 \in \mathscr{M}(S)$, if $m_1 \sim_{cn} m_2$, then $m_1 \sim_{fc} m_2$.

PROOF. If $m_1 \sim_{cn} m_2$, then a cn-bisimulation R exists, containing a triple $(\rho_1^0, C^0, \rho_2^0)$, where C^0 contains no events and $\rho_i^0(Min(C^0)) = m_i$ for $i = 1,2$. Relation $R' = \{((C,\rho_1), id,(C,\rho_2)) \mid (\rho_1,C,\rho_2) \in R\}$, where id is the identity event isomorphism on C, is an fc-bisimulation. As R' contains $((C^0,\rho_1^0),id,(C^0,\rho_2^0))$, it follows that $m_1 \sim_{fc} m_2$. □

The above implications are strict, as illustrated by the following example.

Example 6. Consider the net in Figure 3. In Example 5 we argued that $s_6 \nsim_{cn} s_8$; however, $s_6 \sim_{fc} s_8$, because, even if they do not generate the same causal net, still they generate isomorphic partial orders of events. On the contrary, $s_1 \nsim_{fc} s_3$ because, even if they generate isomorphic partial orders (and so $s_1 =_{pt} s_3$), the two markings have a different branching structure, as discussed in Example 5. Note that the deadlock place s_7 and the empty marking θ are fc-bisimilar, so also partial-order trace equivalent; however, s_7 and θ are not causal-net bisimilar, and not even causal-trace equivalent. □

Definition 18. (State-sensitive fully-concurrent bisimulation) An fc-bisimulation R is *state-sensitive* if for each triple $((C_1,\rho_1),g,(C_2,\rho_2)) \in R$, the maximal markings have equal size, i.e., $|\rho_1(Max(C_1))| = |\rho_2(Max(C_2))|$. Two markings m_1 and m_2 of N are sfc-bisimilar, denoted by $m_1 \sim_{sfc} m_2$, if there exists a state-sensitive fc-bisimulation R containing a triple $((C_1^0,\rho_1^0),g_0,(C_2^0,\rho_2^0))$, where C_i^0 contains no transitions, g_0 is empty and $\rho_i^0(Min(C_i^0)) = \rho_i^0(Max(C_i^0)) = m_i$ for $i = 1,2$. □

Of course, also the above definition is defined coinductively; as we can prove an analogous of Proposition 6, it follows that $\sim_{sfc}$ is an equivalence relation, too.

Now we prove that for BPP nets $\sim_{sfc}$ coincides with causal-net bisimilarity $\sim_{cn}$.

Theorem 1. (Cn-bisimilarity and sfc-bisimilarity coincide) For each BPP net $N = (S,A,T)$, $m_1 \sim_{cn} m_2$ if and only if $m_1 \sim_{sfc} m_2$.

PROOF. $\Rightarrow$) If $m_1 \sim_{cn} m_2$, then there exists a causal-net bisimulation R such that it contains a triple (ρ_1^0,C^0,ρ_2^0), where C^0 contains no transitions and $\rho_i^0(Min(C^0)) = \rho_i^0(Max(C^0)) = m_i$ for $i = 1,2$. Relation $\mathscr{R} = \{((C,\rho_1),id,(C,\rho_2)) \mid (\rho_1,C,\rho_2) \in R\}$, where id is the identity event isomorphism on C, is a state-sensitive fc-bisimulation. Since $\mathscr{R}$ contains the triple $((C^0,\rho_1^0),id,(C^0,\rho_2^0))$, it follows that $m_1 \sim_{sfc} m_2$.

$\Leftarrow$) (Sketch) If $m_1 \sim_{sfc} m_2$, then there exists a state-sensitive fc-bisimulation $\mathscr{R}$ containing a triple $((C_1^0,\rho_1^0),g_0,(C_2^0,\rho_2^0))$, where C_i^0 contains no transitions, g_0 is empty and $\rho_i^0(Min(C_i^0)) = \rho_i^0(Max(C_i^0)) = m_i$ for $i = 1,2$, with $|m_1| = |m_2|$. Hence, C_1^0 and C_2^0 are isomorphic, where the isomorphism function f_0 is a suitably chosen bijection from $Min(C_1^0)$ to $Min(C_2^0)$.[1]

We build the candidate causal-net bisimulation R inductively, by first including the triple $(\rho_1^0,C_1^0,\rho_2^0 \circ f_0)$; hence, if R is a causal-net bisimulation, then $m_1 \sim_{cn} m_2$.

Since $((C_1^0,\rho_1^0),g_0,(C_2^0,\rho_2^0)) \in \mathscr{R}$ and $\mathscr{R}$ is a state-sensitive fc-bisimulation, if $(C_1^0,\rho_1^0) \xrightarrow{e_1} (C_1,\rho_1)$ with $\rho_1(e_1) = t_1$, then $(C_2^0,\rho_2^0) \xrightarrow{e_2} (C_2,\rho_2)$, with $\rho_2(e_2) = t_2$, and $((C_1,\rho_1),g,(C_2,\rho_2)) \in \mathscr{R}$, where $g = g_0 \cup \{(e_1,e_2)\}$, and also with $|\rho_1(Max(C_1))| = |\rho_2(Max(C_2))|$.

It is necessary that the isomorphism bijection f_0 has been chosen in such a way that $f_0(^\bullet e_1) = {^\bullet e_2}$. Since $|\rho_1^0(Max(C_1^0))| = |\rho_2^0(Max(C_2^0))|$ and also $|\rho_1(Max(C_1))| = |\rho_2(Max(C_2))|$, it is necessary that transitions t_1 and t_2 have the same post-set size; hence, C_1 and C_2 are isomorphic and the bijection f_0 can be extended to bijection f with the pair $\{(e_1,e_2)\}$ and also with a suitably chosen bijection between the post-sets of these two transitions. Hence, we include into R also the triple $(\rho_1,C_1,\rho_2 \circ f)$. Symmetrically, if $\rho_2^0(C_2^0)$ moves first.

By iterating this procedure, we add (possibly unboundedly many) triples to R. It is an easy observation to realize that R is a causal-net bisimulation. □

Remark 2. In general, for P/T nets $\sim_{cn}$ is finer than $\sim_{sfc}$. E.g., consider the nets $N = (\{s_1,s_2,s_3,s_4\},\{a\},\{(s_1 \oplus s_2,a,s_3 \oplus s_4)\})$ and $N' = (\{s_1',s_2',s_3'\},\{a\},\{(s_1',a,s_3')\})$. Of course, $s_1 \oplus s_2 \sim_{sfc} s_1' \oplus s_2'$, but $s_1 \oplus s_2 \nsim_{cn} s_1' \oplus s_2'$. □

[1] The actual choice of f_0 (among the $k!$ different bijections, where $k = |m_1| = |m_2|$) will be driven by the bisimulation game that follows; in the light of Corollary 2, it would map team bisimilar places.

3.3. Deadlock-free BPP nets and Fully-Concurrent Bisimilarity

In this section, we first define a cleaning-up operation on a BPP net N, yielding a net $d(N)$ where all the deadlock places of N are removed. Then, we show that two markings m_1 and m_2 of N are fully-concurrent bisimilar if and only if the markings $d(m_1)$ and $d(m_2)$, obtained by removing all the deadlock places in m_1 and m_2 respectively, are state-sensitive fc-bisimilar in $d(N)$.

Definition 19. (Deadlock-free BPP net) For each BPP net $N = (S,A,T)$, we define its associated *deadlock-free* net $d(N)$ as the tuple $(d(S),A,d(T))$ where

- $d(S) = \{s \in S \mid s^\bullet \neq \emptyset\}$;
- $d(T) = \{d(t) \mid t \in T\}$, where $d(t) = (^\bullet t, l(t), d(t^\bullet))$ and $d(m) \in \mathscr{M}(d(S))$ is the marking obtained from $m \in \mathscr{M}(S)$ by removing all its deadlock places.

A BPP net $N = (S,A,T)$ is *deadlock-free* if all of its places are not a deadlock, i.e., $d(S) = S$ and so $d(T) = T$. □

Formally, given a marking $m \in \mathscr{M}(S)$, we define $d(m)$ as the marking

$$d(m)(s) = \begin{cases} m(s) & \text{if } s \in d(S) \\ 0 & \text{otherwise.} \end{cases}$$

For instance, let us consider the net in Figure 3(c). Then, $d(2 \cdot s_6 \oplus 3 \cdot s_7) = 2 \cdot s_6$, or $d(s_7) = \theta$. Of course, $d(m)$ is a multiset on $d(S)$.

The nets in Figure 1 and 2 are deadlock-free. However, note that a deadlock-free net can terminate its computations, but only by reaching the successful termination θ. E.g., the net in Figure 3(a) is deadlock-free. On the contrary, the net in Figure 3(c) is not deadlock-free because of the presence of the deadlock place s_7, and its associated deadlock-free net is $(\{s_6, s_8\}, \{a\}, \{(s_6, a, \theta), (s_8, a, \theta)\}$.

Proposition 10. (Fc-bisimilarity and sfc-bisimilarity coincide on deadlock-free nets) For each deadlock-free BPP net $N = (S,A,T)$, $m_1 \sim_{fc} m_2$ if and only if $m_1 \sim_{sfc} m_2$.

PROOF. $\Leftarrow$) Of course, a state-sensitive fc- bisimulation is also a fc-bisimulation.

$\Rightarrow$) If there are no deadlock places, an fc-bisimulation must be state sensitive. In fact, if two related markings have a different size, then, since no place is a deadlock and the BPP net transitions have singleton pre-set, they would originate different partial orders of events. □

Proposition 11. Given a BPP net $N = (S,A,T)$ and its associated deadlock-free net $d(N) = (d(S),A,d(T))$, two markings m_1 and m_2 of N are fc-bisimilar if and only if $d(m_1)$ and $d(m_2)$ in $d(N)$ are sfc-bisimilar.

PROOF. $\Rightarrow$) If $m_1 \sim_{fc} m_2$, then there exists an fc-bisimulation $\mathscr{R}$ on N containing a triple $((C_1^0, \rho_1^0), g_0, (C_2^0, \rho_2^0))$, where C_i^0 contains no transitions, g_0 is empty and, moreover, $\rho_i^0(Min(C_i^0)) = \rho_i^0(Max(C_i^0)) = m_i$ for $i = 1,2$.

Relation $R = \{((d(C_1), d(\rho_1)), \hat{g}, (d(C_2), d(\rho_2))) \mid ((C_1, \rho_1), g, (C_2, \rho_2)) \in \mathscr{R}$, such that $d(\rho_i)$ is the restriction of ρ_i on the places of $d(C_i)$, for $i = 1,2$, and $\hat{g}$ is such that $g(e_1) = e_2$ implies $\hat{g}(d(e_1)) = d(e_2)\}$ is an fc-bisimulation on $d(N)$. By Proposition

10, R is actually a state-sensitive fully-concurrent bisimulation on $d(N)$. Note that R contains the triple $((d(C_1^0), d(\rho_1^0)), g_0, (d(C_2^0), d(\rho_2^0)))$ such that $d(\rho_i^0)(Min(d(C_i^0))) = d(\rho_i^0)(Max(d(C_i^0))) = d(m_i)$ for $i = 1, 2$, and so $d(m_1) \sim_{sfc} d(m_2)$.

$\Leftarrow$) If $d(m_1) \sim_{sfc} d(m_2)$, then there exists an sfc-bisimulation R on $d(N)$ containing a triple $((\overline{C}_1^0, \overline{\rho}_1^0), g_0, (\overline{C}_2^0, \overline{\rho}_2^0))$, where $\overline{C}_i^0$ contains no transitions, g_0 is empty and $\overline{\rho}_i^0(Min(\overline{C}_i^0)) = \overline{\rho}_i^0(Max(\overline{C}_i^0)) = d(m_i)$ for $i = 1, 2$.

Relation $\mathscr{R} = \{((C_1, \rho_1), g, (C_2, \rho_2)) \mid (C_i, \rho_i)$ is a process of $N(m_{0_i})$ for some m_{0_i}, for $i = 1, 2$, g is an event isomorphism between C_1 and C_2, $((d(C_1), d(\rho_1)), \hat{g}, (d(C_2), d(\rho_2))) \in R$, such that $d(\rho_i)$ is the restriction of ρ_i on the places of $d(C_i)$, for $i = 1, 2$, and $g(e_1) = e_2$ implies $\hat{g}(d(e_1)) = d(e_2)\}$ is an fc-bisimulation on N. Note that relation $\mathscr{R}$ contains the triple $((C_1^0, \rho_1^0), g_0, (C_2^0, \rho_2^0))$ such that, for $i = 1, 2$, $d(C_i^0) = \overline{C}_i^0$, $d(\rho_i^0) = \overline{\rho}_i^0$, $\rho_i^0(Min(C_i^0)) = \rho_i^0(Max(C_i^0)) = m_i$, and therefore $m_1 \sim_{fc} m_2$. □

4. Team Bisimulation Equivalence

In this section, we recall the main definitions and results about team bisimulation equivalence, outlined in [30]. We also include one novel, main result: causal-net bisimilarity coincides with team bisimilarity.

4.1. Additive Closure and its Properties

Definition 20. (Additive closure) Given a BPP net $N = (S, A, T)$ and a *place relation* $R \subseteq S \times S$, we define a *marking relation* $R^\oplus \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$, called the *additive closure* of R, as the least relation induced by the following axiom and rule.

$$\frac{}{(\theta, \theta) \in R^\oplus} \qquad \frac{(s_1, s_2) \in R \quad (m_1, m_2) \in R^\oplus}{(s_1 \oplus m_1, s_2 \oplus m_2) \in R^\oplus}$$

□

Two markings are related by $R^\oplus$ only if they have the same size; in fact, the axiom states that the empty marking is related to itself, while the rule, assuming by induction that m_1 and m_2 have the same size, ensures that $s_1 \oplus m_1$ and $s_2 \oplus m_2$ have the same size.

Proposition 12. For any BPP net $N = (S, A, T)$ and any place relation $R \subseteq S \times S$, if $(m_1, m_2) \in R^\oplus$, then $|m_1| = |m_2|$. □

An alternative way to define that two markings m_1 and m_2 are related by $R^\oplus$ is to state that m_1 can be represented as $s_1 \oplus s_2 \oplus \ldots \oplus s_k$, m_2 can be represented as $s'_1 \oplus s'_2 \oplus \ldots \oplus s'_k$ and $(s_i, s'_i) \in R$ for $i = 1, \ldots, k$.

Proposition 13. For any BPP net $N = (S, A, T)$ and any place relation $R \subseteq S \times S$, the following hold:

1. If R is an equivalence relation, then $R^\oplus$ is an equivalence.
2. If $R_1 \subseteq R_2$, then $R_1^\oplus \subseteq R_2^\oplus$, i.e., the additive closure is monotone.
3. If $(m_1, m_2) \in R^\oplus$ and $(m'_1, m'_2) \in R^\oplus$, then $(m_1 \oplus m'_1, m_2 \oplus m'_2) \in R^\oplus$, i.e., the additive closure is additive.

4. If R is an equivalence relation, $(m_1 \oplus m_1', m_2 \oplus m_2') \in R^{\oplus}$ and $(m_1, m_2) \in R^{\oplus}$, then $(m_1', m_2') \in R^{\oplus}$, i.e., the additive closure is subtractive. □

Example 7. The requirement that R is an equivalence relation is strictly necessary for Propositon 13(4). As a counterexample, consider $R = \{(s_1, s_3), (s_1, s_4), (s_2, s_4)\}$. We have that $(s_1 \oplus s_2, s_3 \oplus s_4) \in R^{\oplus}$ and $(s_1, s_4) \in R^{\oplus}$, but $(s_2, s_3) \notin R^{\oplus}$. □

Remark 3. **(Complexity of additive closure)** Given an equivalence place relation R, the algorithm in [30] checks whether two markings m_1 and m_2 are related by the additive closure of R in $O(k^2)$ time, where k is the size of the markings. In fact, if R is implemented as an adjacency matrix, then the complexity of checking if two markings m_1 and m_2 (represented as an array of places with multiplicities) are related by $R^{\oplus}$ is $O(k^2)$, because the problem is essentially that of finding for each element s_1 of m_1 a matching, R-related element s_2 of m_2. Note that the algorithm in [30] (as well as the similar Algorithm 1 outlined in Section 5) is correct only if R is an equivalence relation, so that $R^{\oplus}$ is subtractive. In fact, assuming that $(m_1, m_2) \in R^{\oplus}$, when we match one place, say s_1, in m_1 with one place, say s_2, in m_2 such that $(s_1, s_2) \in R$, then we need that also $(m_1 \ominus s_1, m_2 \ominus s_2) \in R^{\oplus}$ (*cf.* Example 7).

In general, i.e., for a place relation R that is not an equivalence, the problem of checking whether $(m_1, m_2) \in R^{\oplus}$ has polynomial time complexity because it can be considered as an instance of the problem of finding a perfect matching in a bipartite graph, where the nodes of the two partitions are the tokens in the two markings, and the edges are defined by the relation R. In fact, the definition of the bipartite graph takes $O(k^2)$ time (where $k = |m_1| = |m_2|$) and, then, the Hopcroft-Karp-Karzanov algorithm [34] for computing the maximum matching has worst-case time complexity $O(h\sqrt{k})$, where h is the number of the edges in the bipartire graph ($h \leq k^2$) and to check whether the maximum matching is perfect can be done simply by checking that the size of the matching equals the number of nodes in each partition, i.e., k. □

Now we list some useful, and less obvious, properties of additively closed place relations (proof in [30]).

Proposition 14. For any BPP net $N = (S, A, T)$ and any family of place relations $R_i \subseteq S \times S$, the following hold:

1. $\emptyset^{\oplus} = \{(\theta, \theta)\}$, i.e., the additive closure of the empty place relation is a singleton marking relation, relating the empty marking to itself.
2. $(\mathscr{I}_S)^{\oplus} = \mathscr{I}_M$, i.e., the additive closure of the identity relation on places $\mathscr{I}_S = \{(s, s) \mid s \in S\}$ is the identity relation on markings $\mathscr{I}_M = \{(m, m) \mid m \in \mathscr{M}(S)\}$.
3. $(R^{\oplus})^{-1} = (R^{-1})^{\oplus}$, i.e., the inverse of an additively closed relation R is the additive closure of its inverse R^{-1}.
4. $(R_1 \circ R_2)^{\oplus} = (R_1^{\oplus}) \circ (R_2^{\oplus})$, i.e., the additive closure of the composition of two place relations is the compositions of their additive closures.
5. $\bigcup_{i \in I}(R_i^{\oplus}) \subseteq (\bigcup_{i \in I} R_i)^{\oplus}$, i.e., the union of additively closed relations is included into the additive closure of their union. □

4.2. Team Bisimulation on Places

Definition 21. (Team bisimulation) Let $N = (S, A, T)$ be a BPP net. A *team bisimulation* is a place relation $R \subseteq S \times S$ such that if $(s_1, s_2) \in R$ then for all $\ell \in A$

- $\forall m_1$ such that $s_1 \xrightarrow{\ell} m_1$, $\exists m_2$ such that $s_2 \xrightarrow{\ell} m_2$ and $(m_1, m_2) \in R^{\oplus}$,
- $\forall m_2$ such that $s_2 \xrightarrow{\ell} m_2$, $\exists m_1$ such that $s_1 \xrightarrow{\ell} m_1$ and $(m_1, m_2) \in R^{\oplus}$.

Two places s and s' are *team bisimilar* (or *team bisimulation equivalent*), denoted $s \sim s'$, if there exists a team bisimulation R such that $(s, s') \in R$. □

Example 8. Continuing Example 1 about the semi-counters in Figure 1, it is easy to see that relation $R = \{(s_1, s_3), (s_1, s_4), (s_2, s_5), (s_2, s_6)\}$ is a team bisimulation. In fact, the pair (s_1, s_3) is a team bisimulation pair because, to transition $s_1 \xrightarrow{inc} s_1 \oplus s_2$, s_3 can respond with $s_3 \xrightarrow{inc} s_4 \oplus s_5$, and $(s_1 \oplus s_2, s_4 \oplus s_5) \in R^{\oplus}$; symmetrically, if s_3 moves first. Also the pair (s_1, s_4) is a team bisimulation pair because, to transition $s_1 \xrightarrow{inc} s_1 \oplus s_2$, s_4 can respond with $s_4 \xrightarrow{inc} s_3 \oplus s_6$, and $(s_1 \oplus s_2, s_3 \oplus s_6) \in R^{\oplus}$; symmetrically, if s_4 moves first. Also the pair (s_2, s_5) is a team bisimulation pair: to transition $s_2 \xrightarrow{dec} \theta$, s_5 responds with $s_5 \xrightarrow{dec} \theta$, and $(\theta, \theta) \in R^{\oplus}$. Similarly for the pair (s_2, s_6). Therefore, relation R is a team bisimulation, indeed. This example shows that $\sim$ is compatible with the notion of net unfolding, as the net (b) can be seen as a sort of partial unfolding of the net (a).

The team bisimulation R above is a very simple, finite relation, proving that s_1 and s_3 are team bisimulation equivalent. In Example 2, in order to show that s_1 and s_3 are interleaving bisimilar, we had to introduce a complex relation, with infinitely many pairs. In Example 3 we argued that $s_1 \sim_{cn} s_3$, even if we did not provide any causal-net bisimulation relation (which would be composed of infinitely many triples). □

Example 9. Let us consider the nets in Figure 2. Of course, $s_1 \sim s_2$ because
$R = \{(s_1, s_2), (s_1, s_3), (s_1, s_4)\}$
is a team bisimulation. Actually, all the places are pairwise team bisimilar. In Example 4 we argued that $s_1 \sim_{cn} s_2$, but the justifying causal-net bisimulation would contain infinitely many triples. □

Example 10. Consider the nets in Figure 4. It is easy to see that $R = \{(s_1, s_4), (s_2, s_5), (s_2, s_6), (s_2, s_7), (s_3, s_8), (s_3, s_9)\}$ is a team bisimulation. This example shows that team bisimulation is compatible with duplication of behavior and fusion of places. □

We now list some useful properties of team bisimulation relations.

Proposition 15. For any BPP net $N = (S, A, T)$, the following hold:

1. The identity relation $\mathscr{I}_S = \{(s, s) \mid s \in S\}$ is a team bisimulation;
2. the inverse relation $R^{-1} = \{(s', s) \mid (s, s') \in R\}$ of a team bisimulation R is a team bisimulation;
3. the relational composition $R_1 \circ R_2 = \{(s, s'') \mid \exists s'.(s, s') \in R_1 \wedge (s', s'') \in R_2\}$ of two team bisimulations R_1 and R_2 is a team bisimulation;

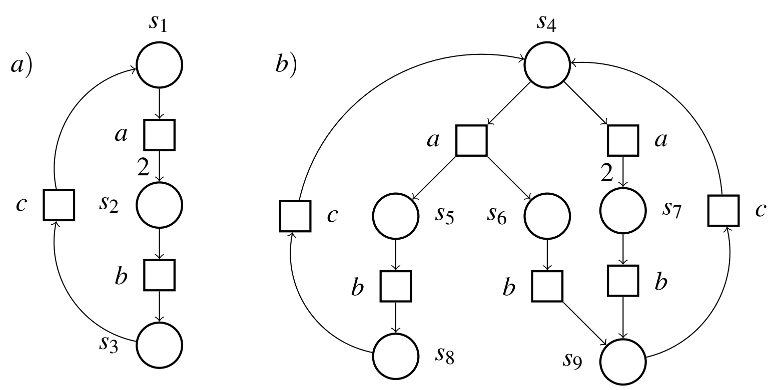

Figure 4: Two team bisimilar BPP nets

4. the union $\bigcup_{i\in I} R_i$ of team bisimulations R_i is a team bisimulation.

PROOF. Standard, by exploiting Proposition 14; details in [30]. □

Remember that $s \sim s'$ if there exists a team bisimulation containing the pair (s,s'). This means that $\sim$ is the union of all team bisimulations, i.e.,

$$\sim = \bigcup \{R \subseteq S \times S \mid R \text{ is a team bisimulation}\}.$$

By Proposition 15(4), $\sim$ is also a team bisimulation, hence the largest such relation.

Proposition 16. For any BPP net $N = (S,A,T)$, relation $\sim \subseteq S \times S$ is the largest team bisimulation relation. □

Proposition 17. For any BPP net $N = (S,A,T)$, relation $\sim \subseteq S \times S$ is an equivalence relation.

PROOF. By Proposition 15; details in [30]. □

Remark 4. (Complexity of $\sim$) The well-known Kanellakis-Smolka algorithm for computing bisimulation equivalence over a finite-state LTS with n states and m transitions has $O(m \cdot n)$ time complexity [37, 38]. This very same partition refinement algorithm can be easily adapted also for team bisimilarity over BPP nets: it is enough to consider the empty marking θ as an additional, special place which is team bisimilar to itself only (i.e., the initial partition is composed of two blocks: S and $\{\theta\}$), and to consider the little additional cost due to the fact that the reached markings are to be related by the additive closure of the current partition over places; this extra cost is related to the size of the post-set of the net transitions; if p is the size of the largest post-set of the net transitions (i.e., p is the least number such that $|t^\bullet| \leq p$, for all $t \in T$), then the time complexity is $O(m \cdot p^2 \cdot n)$, where m is the number of the net transitions and n is the number of the net places. □

4.3. Team Bisimilarity over Markings

Starting from team bisimulation equivalence $\sim$, which has been computed over the places of an *unmarked* BPP net N, we can lift it over *the markings* of N in a distributed way: m_1 is team bisimulation equivalent to m_2 if these two markings are related by the additive closure of $\sim$, i.e., if $(m_1, m_2) \in \sim^{\oplus}$, usually denoted by $m_1 \sim^{\oplus} m_2$.

Proposition 18. For any BPP net $N = (S, A, T)$, if $m_1 \sim^{\oplus} m_2$, then $|m_1| = |m_2|$.

PROOF. By Proposition 12. □

Proposition 19. For any BPP net $N = (S, A, T)$, relation $\sim^{\oplus} \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$ is an equivalence relation.

PROOF. By Proposition 13: since $\sim$ is an equivalence relation (Proposition 17), its additive closure $\sim^{\oplus}$ is also an equivalence relation (Proposition 13(1)). □

Remark 5. (Complexity of $\sim^{\oplus}$) Once the place relation $\sim$ has been computed for the given net (in $O(m \cdot p^2 \cdot n)$ time), by using the algorithm in [30] we can check whether two markings m_1 and m_2 are team bisimulation equivalent in $O(k^2)$ time, where k is the size of the markings, as discussed in Remark 3. Moreover, if we want to check whether other two markings of the same net are team bisimilar, we can reuse the already computed $\sim$ relation, so that the time complexity is again quadratic on the size of the two markings. However, note that the time spent in creating the adjacency matrix for $\sim$ has not been considered: since n is the number of places, $O(n^2)$ time is needed to implement this matrix, so that the time spent for the first check is $O(n^2)$, while for subsequent checks it is only $O(k^2)$, where k is the size of the markings.

The algorithm in [30] is not optimal. As a matter of fact, since the partition refinement algorithm does compute the equivalence classes of $\sim$, we can take advantage of this fact for checking whether $m_1 \sim^{\oplus} m_2$. The algorithm in [41] simply scans these equivalence classes and, for each class, it checks whether the number of tokens in the places of m_1 belonging to this class equals the number of tokens in the places of m_2 in the same class; if this holds for all the equivalence classes, then $m_1 \sim^{\oplus} m_2$. Of course, the complexity of this algorithm is $O(n)$, even for the first check; hence, this algorithm is usually more performant, even if, from the second check onwards, it may be slower when applied to small markings; in fact, in case the number n of places is greater than k^2, then the original algorithm is better. This more performant algorithm can be used in computing $\sim$ when it is necessary to check that the reached markings are to be related by the additive closure of the equivalence relation induced by the current partition over places. In such a case, the complexity of $\sim$ is $O(m \cdot n^2)$. □

Example 11. Continuing Example 8 about the semi-counters, the marking $s_1 \oplus 2 \cdot s_2$ is team bisimilar to the following markings of the net in (b): $s_3 \oplus 2 \cdot s_5$, or $s_3 \oplus s_5 \oplus s_6$, or $s_3 \oplus 2 \cdot s_6$, or $s_4 \oplus 2 \cdot s_5$, or $s_4 \oplus s_5 \oplus s_6$, or $s_4 \oplus 2 \cdot s_6$. □

Of course, two markings m_1 and m_2 are *not* team bisimilar if there is no bijective, team-bisimilar-preserving mapping between them; this is the case when m_1 and m_2 have different size, or if the algorithm in [30] ends with b holding *false*, i.e., by singling out a place s_i' in (the residual of) m_1 which has no matching team bisimilar place in (the residual of) m_2.

The following theorem provides a characterization of team bisimulation equivalence $\sim^{\oplus}$ as a suitable bisimulation-like relation over markings. It is interesting to observe that this characterization gives a dynamic interpretation of team bisimulation equivalence, while Definition 20 gives a structural definition of team bisimulation equivalence $\sim^{\oplus}$ as the additive closure of $\sim$. The proof is outlined in [30].

Theorem 2. Let $N = (S, A, T)$ be a BPP net. Two markings m_1 and m_2 are team bisimulation equivalent, $m_1 \sim^{\oplus} m_2$, if and only if $|m_1| = |m_2|$ and

- $\forall t_1$ such that $m_1[t_1\rangle m_1'$, $\exists t_2$ such that $^\bullet t_1 \sim {}^\bullet t_2$, $l(t_1) = l(t_2)$, $t_1^\bullet \sim^{\oplus} t_2^\bullet$, $m_2[t_2\rangle m_2'$ and $m_1' \sim^{\oplus} m_2'$, and symmetrically,
- $\forall t_2$ such that $m_2[t_2\rangle m_2'$, $\exists t_1$ such that $^\bullet t_1 \sim {}^\bullet t_2$, $l(t_1) = l(t_2)$, $t_1^\bullet \sim^{\oplus} t_2^\bullet$, $m_1[t_1\rangle m_1'$ and $m_1' \sim^{\oplus} m_2'$. □

By the theorem above, it is clear that $\sim^{\oplus}$ is an interleaving bisimulation.

Corollary 1. **(Team bisimilarity is finer than interleaving bisimilarity)** Let $N = (S, A, T)$ be a BPP net. If $m_1 \sim^{\oplus} m_2$, then $m_1 \sim_{int} m_2$. □

4.4. Team Bisimilarity and Causal-net Bisimilarity Coincide

Now we want to prove an original result: team bisimilarity coincides with cn-bisimilarity (see Definition 16), hence proving that team bisimilarity does respect also the causal semantics of BPP nets.

Theorem 3. **(Team bisimilarity implies cn-bisimilarity)** Let $N = (S, A, T)$ be a BPP net. If $m_1 \sim^{\oplus} m_2$, then $m_1 \sim_{cn} m_2$.

PROOF. Let $R = \{(\rho_1, C, \rho_2) \mid (C, \rho_1)$ is a process of $N(m_1)$ and (C, ρ_2) is a process of $N(m_2)$ such that $\rho_1(s) \sim \rho_2(s)$, for all $s \in Max(C)\}$. We want to prove that R is a causal-net bisimulation.

First, consider a triple of the form $(\rho_1^0, C^0, \rho_2^0)$, where C^0 is a BPP causal net with no transitions, $\rho_i^0(Max(C^0)) = m_i$ (for $i = 1, 2$) and $\rho_1^0(s) \sim \rho_2^0(s)$, for all $s \in Max(C^0)$. Then $(\rho_1^0, C^0, \rho_2^0)$ must belong to R, because (C^0, ρ_i^0) is a process of $N(m_i)$, for $i = 1, 2$ and, by hypothesis, $(m_1, m_2) \in R_1^{\oplus}$. Note also that if the relation R is a causal-net bisimulation, then this triple ensures that $m_1 \sim_{cn} m_2$.

Now assume $(\rho_1, C, \rho_2) \in R$. In order to be a causal-net bisimulation triple, it is necessary that

i) $\forall t_1, C', \rho_1'$ such that $(C, \rho_1) \xrightarrow{e} (C', \rho_1')$, with $\rho_1'(e) = t_1$, there exist t_2, ρ_2' such that $(C, \rho_2) \xrightarrow{e} (C', \rho_2')$, with $\rho_2'(e) = t_2$, and $(\rho_1', C', \rho_2') \in R$;

ii) symmetrically, if (C, ρ_2) moves first.

Let t_1 be a transition such that $(C, \rho_1) \xrightarrow{e} (C', \rho_1')$, with $\rho_1'(e) = t_1$, and let $s_1 = {}^\bullet t_1$. As by hypothesis we have that $\rho_1(s) \sim \rho_2(s)$, for all $s \in Max(C)$, if $s_1 = \rho_1(s')$, then there exists $s_2 = \rho_2(s')$ such that $s_1 \sim s_2$. Hence, t_2 exists such that $s_1 = {}^\bullet t_1 \sim {}^\bullet t_2 = s_2$, $l(t_1) = l(t_2)$, $t_1^\bullet \sim^\oplus t_2^\bullet$, so that, by Theorem 2, $\rho_2(Max(C))[t_2\rangle m_2'$ and $m_1' \sim^\oplus m_2'$. Hence, it is really possible to extend the causal net C to the causal net C' through a suitable transition e such that ${}^\bullet e = s'$, as required above, and to extend ρ_1 and ρ_2 to ρ_1' and ρ_2', respectively, in such a way that $\rho_1'(e) = t_1$, $\rho_2'(e) = t_2$, and $\rho_1'(s) \sim \rho_2'(s)$, for all $s \in t^\bullet$ because $t_1^\bullet \sim^\oplus t_2^\bullet$. Finally, we have that $(\rho_1', C', \rho_2') \in R$ because $\rho_1'(s) \sim \rho_2'(s)$, for all $s \in Max(C')$, as required. Symmetrically, if (C, ρ_2) moves first. □

Theorem 4. **(Cn-bisimilarity implies team bisimilarity)** Let $N = (S, A, T)$ be a BPP net. If $m_1 \sim_{cn} m_2$ then $m_1 \sim^\oplus m_2$.

PROOF. If $m_1 \sim_{cn} m_2$, then there exists a causal-net bisimulation R containing a triple $(\rho_1^0, C^0, \rho_2^0)$, where C^0 is a BPP causal net which has no transitions and $\rho_i^0(Max(C^0)) = m_i$ for $i = 1, 2$. Let us consider $\mathscr{R} = \{(\rho_1(s), \rho_2(s)) \mid (\rho_1, C, \rho_2) \in R \wedge s \in Max(C)\}$. If we prove that $\mathscr{R}$ is a team bisimulation, then, since $(\rho_1^0(s), \rho_2^0(s)) \in \mathscr{R}$ for each $s \in Max(C^0)$, it follows that $(m_1, m_2) \in \mathscr{R}^\oplus$. As $\mathscr{R} \subseteq \sim$, we also get $m_1 \sim^\oplus m_2$. Let us consider a pair $(s_1, s_2) \in \mathscr{R}$. Hence, there exist a triple $(\rho_1, C, \rho_2) \in R$ and a place $s \in Max(C)$ such that $s_1 = \rho_1(s)$ and $s_2 = \rho_2(s)$. If s_1 moves, e.g., $t_1 = s_1 \xrightarrow{\ell} m_1'$, then $\rho_1(Max(C))[t_1\rangle \overline{m}_1$, where $\overline{m}_1 = \rho_1(Max(C)) \ominus s_1 \oplus m_1'$. Hence, also $(C, \rho_1) \xrightarrow{e} (C', \rho_1')$, with $\rho_1'(e) = t_1$, for some (C', ρ_1').

Since R is a causal-net bisimulation, there exist t_2, ρ_2' such that $(C, \rho_2) \xrightarrow{e} (C', \rho_2')$, with $\rho_2'(e) = t_2$, and $(\rho_1', C', \rho_2') \in R$.

Note that event e is such that ${}^\bullet e = s$, and so ${}^\bullet t_2 = s_2$. This means that $\overline{m}_2 = \rho_2(Max(C)) \ominus s_2 \oplus m_2'$, where $m_2' = t_2^\bullet$; in other words, $t_2 = s_2 \xrightarrow{\ell} m_2'$. Note also that ρ_1' extends ρ_1 by mapping e to t_1 and, similarly, ρ_2' extends ρ_2 by mapping e to t_2; in this way, $\rho_1'(e^\bullet) = t_1^\bullet$ and $\rho_2'(e^\bullet) = t_2^\bullet$. Since $(\rho_1', C', \rho_2') \in R$, it follows that the set $\{(\rho_1'(s'), \rho_2'(s')) \mid s' \in e^\bullet\}$ is a subset of $\mathscr{R}$, so that $(m_1', m_2') \in \mathscr{R}^\oplus$.

Summing up, for each pair $(s_1, s_2) \in \mathscr{R}$, if $s_1 \xrightarrow{\ell} m_1'$, then $s_2 \xrightarrow{\ell} m_2'$ such that $(m_1', m_2') \in \mathscr{R}^\oplus$; symmetrically, if s_2 moves first. So, $\mathscr{R}$ is a team bisimulation. □

Corollary 2. **(Team bisimilarity and cn-bisimilarity coincide)** Let $N = (S, A, T)$ be a BPP net. Then, $m_1 \sim_{cn} m_2$ if and only if $m_1 \sim^\oplus m_2$.

PROOF. By Theorems 3 and 4, we get the thesis. □

Corollary 3. **(Team bisimilarity and sfc-bisimilarity coincide)** Let $N = (S, A, T)$ be a BPP net. Then, $m_1 \sim_{sfc} m_2$ if and only if $m_1 \sim^\oplus m_2$.

PROOF. By Corollary 2 and Theorem 1, we get the thesis. □

Therefore, our characterization of cn-bisimilarity and sfc-bisimilarity, which are, in our opinion, the intuitively correct (strong) causal semantics for BPP nets, is quite appealing because it is based on the very simple technical definition of team bisimulation on the places of the unmarked net, and, moreover, offers a very efficient algorithm to check if two markings are cn-bisimilar (see Remark 5).

5. H-Team Bisimulation

In order to provide the definition of *h-team bisimulation on places* for unmarked BPP nets, adapting the definition of team bisimulation on places (cf. Definition 21), we need first to extend the domain of a place relation: the empty marking θ is considered as an additional place, so that a place relation is defined not on S, rather on $S \cup \{\theta\}$. Therefore, the symbols r_1 and r_2 that occur in the following definitions, can only denote either the empty marking θ or a single place s.

First of all, we extend the idea of additive closure to these more general place relations, yielding *h-additive closure*.

Definition 22. (H-additive closure) Given a BPP net $N = (S, A, T)$ and a *place relation* $R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$, we define a *marking relation* $R^\oplus \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$, called the *h-additive closure* of R, as the least relation induced by the following axiom and rule.

$$\frac{}{(\theta, \theta) \in R^\oplus} \qquad \frac{(r_1, r_2) \in R \quad (m_1, m_2) \in R^\oplus}{(r_1 \oplus m_1, r_2 \oplus m_2) \in R^\oplus}$$

□

Note that if two markings are related by $R^\oplus$ (i.e., by the h-additive closure of R), then they may have different size; in fact, even if the axiom relates the empty marking to itself (so two markings with the same size), as $R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$, it may be the case that $(\theta, s) \in R$, so that, assuming $(m_1', m_2') \in R^\oplus$ with $|m_1'| = |m_2'|$, we get $(m_1', s \oplus m_2') \in R^\oplus$, as θ is the identity for the operator of multiset union. Hence, Proposition 12, which is valid for place relations defined over S, is not valid for place relations defined over $S \cup \{\theta\}$. However, the properties in Propositions 13 and 14 hold also for these more general place relations. In particular, if $R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$ is an equivalence relation, then $R^\oplus$ is also an equivalence relation.

Proposition 20. (H-additivity/H-subtractivity) Given a BPP net $N = (S, A, T)$ and a place relation $R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$, the following hold:

1. If $(m_1, m_2) \in R^\oplus$ and $(m_1', m_2') \in R^\oplus$, then $(m_1 \oplus m_1', m_2 \oplus m_2') \in R^\oplus$.
2. If R is an equivalence relation, $(m_1 \oplus m_1', m_2 \oplus m_2') \in R^\oplus$ and $(m_1, m_2) \in R^\oplus$, then $(m_1', m_2') \in R^\oplus$.

PROOF. By induction on the size of m_1. This easy inductive proof is very similar to the analogous one in [30] for Proposition 13 and so omitted. □

Remark 6. (Complexity of h-additive closure) Given an equivalence place relation $R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$, the complexity of checking if two markings m_1 and m_2 are related by $R^\oplus$ is $O(k^2)$, where k is the size of the largest marking, since the problem is essentially that of finding for each element s_1 (not R-related to θ) of m_1 a matching, R-related element s_2 of m_2 (and then checking that all the remaining elements of m_1 and m_2 are R-related to θ), as described by Algorithm 1. Note that this algorithm returns *false* in case m_1 and m_2 are not related by $R^\oplus$, otherwise it returns the set P of matched pairs (of places, or composed of a place and the empty marking θ) which are necessary to prove that m_1 and m_2 are related by $R^\oplus$. Moreover, note that this algorithm is correct only if R is an equivalence relation, so that $R^\oplus$ is h-subtractive (*cf.* Remark 3). □

Algorithm 1 Checking the H-additive Closure of an Equivalence Place Relation

Let $N = (S, A, T)$ be *BPP* net.
Let $R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$ be a place relation, which is an equivalence.
Let A be the adjacency matrix generated as follows: $A[r, r'] = 1$ if $(r, r') \in R$; otherwise $A[r, r'] = 0$, where r, r' range over $S \cup \{\theta\}$.
Let $m_1 = k_1 \cdot s_{11} \oplus k_2 \cdot s_{12} \oplus \ldots \oplus k_{j_1} \cdot s_{1j_1}$ such that $k_i > 0$ for $i = 1, \ldots, j_1$. Let M_1 be an array of length j_1 such that $M_1[j] = k_j$, for $j = 1, \ldots, j_1$.
Let $m_2 = h_1 \cdot s_{21} \oplus h_2 \cdot s_{22} \oplus \ldots \oplus h_{j_2} \cdot s_{2j_2}$ such that $h_i > 0$ for $i = 1, \ldots, j_2$. Let M_2 be an array of length j_2 such that $M_2[j] = h_j$, for $j = 1, \ldots, j_2$.

```
Let P be the set of currently matched R-related places, initialized to ∅
for i = 1 to j_1 do
    if A[s_1i, θ] == 1 then
        add (s_1i, θ) to P
    else
        for j = 1 to M_1[i] do
            h = 1
            b = true
            while (h ≤ j_2 and b) do
                if M_2[h] ≠ 0   and   A[s_1i, s_2h] == 1 then
                    add (s_1i, s_2h) to P
                    M_2[h] = M_2[h] − 1
                    b = false
                else
                    h = h + 1
                end if
            end while
            if h > j_2 then
                return false
            end if
        end for
    end if
end for
for i = 1 to j_2 do
    if M_2[i] > 0 then
        if A[θ, s_2i] == 0 then
            return false
        else
            add (θ, s_2i) to P
        end if
    end if
end for
return P
```

Now we are ready to define *h-team bisimulation*, where the symbols r_1 and r_2 can only denote either the empty marking θ or a single place, because of the shape of BPP net transitions.

Definition 23. (H-team bisimulation) Let $N = (S, A, T)$ be a BPP net. An *h-team bisimulation* is a place relation $R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$ such that if $(r_1, r_2) \in R$ then for all $\ell \in A$

- $\forall m_1$ such that $r_1 \xrightarrow{\ell} m_1$, $\exists m_2$ such that $r_2 \xrightarrow{\ell} m_2$ and $(m_1, m_2) \in R^{\oplus}$,
- $\forall m_2$ such that $r_2 \xrightarrow{\ell} m_2$, $\exists m_1$ such that $r_1 \xrightarrow{\ell} m_1$ and $(m_1, m_2) \in R^{\oplus}$.

r_1 and r_2 are *h-team bisimilar* (or *h-team bisimulation equivalent*), denoted $r_1 \sim_h r_2$, if there exists an h-team bisimulation R such that $(r_1, r_2) \in R$. □

Since a team bisimulation is also an h-team bisimulation, we have that $\sim$ implies $\sim_h$. This implication is strict as illustrated in the following examples.

Example 12. Consider the nets in Figure 3. It is not difficult to realize that s_6 and s_8 are h-team bisimilar because $R = \{(s_6, s_8), (s_7, \theta)\}$ is a h-team bisimulation. In fact, s_6 can reach s_7 by performing a, and s_8 can reply by reaching the empty marking θ, and $(s_7, \theta) \in R$. In Example 6 we argued that $s_6 \sim_{fc} s_8$ and in fact we will prove that h-team bisimilarity coincide with fc-bisimilarity. This example shows that h-team bisimulation equivalence is not sensitive to the kind of termination of a process: even if s_7 is stuck place, denoting a deadlock situation, it is equivalent to the empty marking θ, i.e., the marking denoting a properly terminated process. This is in contrast with the definition of team bisimulation on place, which is sensitive to the kind of termination. In fact, $s_6 \nsim s_8$, because $s_6 \nsim_{cn} s_8$, as discussed in Example 5. □

Remark 7. (Only deadlock places can be related to θ) It is interesting to observe that an h-team bisimulation can relate θ only to deadlock places, i.e., places with empty post-set. This is indeed the case because a place s of a BPP net such that $s^\bullet \neq \emptyset$ can perform some transition (as net transitions of a BPP net have singleton pre-set), while θ can do nothing; hence, (s, θ) cannot be an h-team bisimulation pair. □

We now list some useful properties which can be proved similarly to Proposition 15 (whose proof is in [30]).

Proposition 21. For any BPP net $N = (S, A, T)$, the following hold:

1. The identity relation $\mathscr{I}_S = \{(r, r) \mid r \in S \cup \{\theta\}\}$ is an h-team bisimulation;
2. the inverse relation $R^{-1} = \{(r', r) \mid (r, r') \in R\}$ of an h-team bisimulation R is an h-team bisimulation;
3. the relational composition $R_1 \circ R_2 = \{(r, r'') \mid \exists r'.(r, r') \in R_1 \wedge (r', r'') \in R_2\}$ of two h-team bisimulations R_1 and R_2 is an h-team bisimulation;
4. the union $\bigcup_{i \in I} R_i$ of h-team bisimulations R_i is an h-team bisimulation. □

Relation $\sim_h$ is the union of all h-team bisimulations, i.e.,

$$\sim_h = \bigcup \{R \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\}) \mid R \text{ is an h-team bisimulation}\}.$$

By Proposition 21(4), $\sim_h$ is also an h-team bisimulation, hence the largest such relation.

Proposition 22. For any BPP net $N = (S,A,T)$, relation $\sim_h \subseteq (S \cup \{\theta\}) \times (S \cup \{\theta\})$ is an equivalence relation.

PROOF. Standard, by Proposition 21. □

Moreover, h-team bisimulation on places enjoys the same properties of bisimulation on LTSs, i.e., it is coinductive and equipped with a fixed-point characterization. H-team bisimulation equivalence can be used in order to minimize the net in the style of [30]. In particular, the minimized net $N_{\sim_h}$ w.r.t. $\sim_h$ of a BPP net N is the deadlock-free net $d(N_\sim)$, where $N_\sim$ is the minimized net w.r.t. $\sim$, as described in [30].

Starting from h-team bisimulation equivalence $\sim_h$, which has been computed over the places (and the empty marking) of an *unmarked* BPP net N, we can lift it over *the markings* of N in a distributed way: m_1 is h-team bisimulation equivalent to m_2 if these two markings are related by the h-additive closure of $\sim_h$, i.e., if $(m_1, m_2) \in \sim_h^\oplus$, usually denoted by $m_1 \sim_h^\oplus m_2$.

Proposition 23. For any BPP net $N = (S,A,T)$, relation $\sim_h^\oplus \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$ is an equivalence relation.

PROOF. By (the analogous of) Proposition 13 (for h-additive closure): since $\sim_h$ is an equivalence relation (Proposition 22), its h-additive closure $\sim_h^\oplus$ is also an equivalence relation. □

Remark 8. (Complexity of $\sim_h^\oplus$) Computing $\sim_h$ is not more difficult than computing $\sim$ (cf. Remark 5). The partition refinement algorithm in [37, 38] can be adapted also to this case. It is enough to consider the empty marking θ as an additional, special place which is h-team bisimilar to each deadlock place. Hence, the initial partition considers two sets: one composed of all the deadlock places and θ, the other one with all the non-deadlock places. Therefore, the time complexity is also in this case $O(m \cdot p^2 \cdot n)$, where m is the number of the net transitions, n is the number of the net places and p the size of the largest post-set of the net transitions.

Once the equivalence place relation $\sim_h$ has been computed once and for all for the given net, by using the Algorithm 1 we can check whether two markings m_1 and m_2 are h-team bisimulation equivalent in $O(k^2)$ time, where k is the size of the largest marking, as discussed in Remark 6.

Since the partition refinement algorithm does compute the equivalence classes of $\sim_h$, an alternative algorithm [41] for checking whether $m_1 \sim_h^\oplus m_2$ simply scans these equivalence classes and, for each class (except for the class of θ), it checks whether the number of tokens in the places of m_1 belonging to this class equals the number of tokens in the places of m_2 in the same class; if this holds for all the equivalence classes, then $m_1 \sim_h^\oplus m_2$. Of course, the complexity of this algorithm is $O(n)$. Again, this more performant algorithm can be used in computing $\sim_h$ so that the complexity of $\sim_h$ in this case becomes $O(m \cdot n^2)$ time. □

5.1. H-team Bisimilarity and Fully-concurrent Bisimilarity Coincide

In this section, we first show that h-team bisimilarity over a BPP net N coincides with team-bisimilarity over its associated deadlock-free net $d(N)$. A consequence of this result is that h-team bisimilarity coincides with fc-bisimilarity on BPP nets.

Proposition 24. Given a BPP net $N = (S, A, T)$ and its associated deadlock-free net $d(N) = (d(S), A, d(T))$, two markings m_1 and m_2 of N are h-team bisimilar if and only if $d(m_1)$ and $d(m_2)$ in $d(N)$ are team bisimilar.

PROOF. $\Rightarrow$) If $m_1 \sim_h^\oplus m_2$, then there exists an h-team bisimulation R_1 on N such that $(m_1, m_2) \in R_1^\oplus$. If we take relation $R_2 = \{(s_1, s_2) \mid s_1, s_2 \in d(S) \wedge (s_1, s_2) \in R_1\}$, then it is easy to see that R_2 is a team bisimulation on $d(N)$, so that $(d(m_1), d(m_2)) \in R_2^\oplus$, hence $d(m_1) \sim^\oplus d(m_2)$.

$\Leftarrow$) If $d(m_1) \sim^\oplus d(m_2)$, then there exists a team bisimulation R_2 on $d(N)$ such that $(d(m_1), d(m_2)) \in R_2^\oplus$. Now, take relation $R_1 = R_2 \cup (S' \cup \{\theta\}) \times (S' \cup \{\theta\})$, where the set S' is $\{s \in S \mid s^\bullet = \emptyset\}$. It is easy to observe that R_1 is an h-team bisimulation on N, so that $(m_1, m_2) \in R_1^\oplus$, hence $m_1 \sim_h^\oplus m_2$. □

Theorem 5. (Fully concurrent bisimilarity and h-team bisimilarity coincide) Given a BPP net $N = (S, A, T)$, $m_1 \sim_{fc} m_2$ if and only if $m_1 \sim_h^\oplus m_2$.

PROOF. By Proposition 11, $m_1 \sim_{fc} m_2$ in N if and only if $d(m_1) \sim_{sfc} d(m_2)$ in the associated deadlock-free net $d(N)$. By Corollary 3, $d(m_1) \sim_{sfc} d(m_2)$ if and only if $d(m_1) \sim^\oplus d(m_2)$ in $d(N)$. By Proposition 24, $d(m_1) \sim^\oplus d(m_2)$ in $d(N)$ if and only if $m_1 \sim_h^\oplus m_2$ in N. The thesis then follows by transitivity. □

5.2. Modal Logic Characterization

In this section we extend Hennessy-Milner Logic (HML) [33, 2] with an operator $\otimes$ of parallel composition of formulae. The resulting modal logic, called HTML (H-Team Modal Logic), is a simplification of the modal logic TML (Team Modal Logic), proposed in [30] in order to characterize team bisimulation equivalence for BPP nets. We will prove that HTML model checking is coherent with h-team equivalence checking: two markings are h-team bisimilar if and only if they satisfy the same HTML formulae.

The HTML *formulae* are generated from the finite set A of actions by the following abstract syntax:

$$F \ ::= \ tt \mid ff \mid F \wedge F \mid F \vee F \mid \neg F \mid \langle a \rangle F \mid [a]F \mid F \otimes F$$

where a is any action in A, tt and ff are two atomic proposition (for *true* and *false*, respectively), $\wedge$ is the operator of logical conjunction, $\vee$ is disjunction, $\langle a \rangle F$ denotes *possibility* (it is possible to do a and then reach a marking where F holds), $[a]F$ denotes *necessity* (by doing a, only markings where F holds can be reached), $\neg$ is logical negation and, finally, $\otimes$ is the operator of parallel composition of formulae.

We denote by $\mathscr{F}_A$ the set of all HTML formulae, built from the actions in A. We sometimes use some useful abbreviations: if $B = \{a_1, a_2, \ldots, a_k\} \subseteq A$, $k \geq 1$, then $\langle B \rangle F$ stands for $\langle a_1 \rangle F \vee \langle a_2 \rangle F \vee \ldots \vee \langle a_k \rangle F$, and $[B]F$ stands for $[a_1]F \wedge [a_2]F \wedge \ldots \wedge [a_k]F$.

$$[\![tt]\!] = \mathscr{M}(S) \qquad [\![ff]\!] = \emptyset$$

$$[\![F_1 \wedge F_2]\!] = [\![F_1]\!] \cap [\![F_2]\!] \qquad [\![F_1 \vee F_2]\!] = [\![F_1]\!] \cup [\![F_2]\!] \qquad [\![\neg F]\!] = \mathscr{M}(S) \setminus [\![F]\!]$$

$$[\![\langle a \rangle F]\!] = \{s \in S \mid \exists m. s \xrightarrow{a} m \text{ and } d(m) \in [\![F]\!]\}$$

$$[\![[a]F]\!] = \{s \in S \cup \{\theta\} \mid \forall m (s \xrightarrow{a} m \text{ implies } d(m) \in [\![F]\!])\}$$

$$[\![F_1 \otimes F_2]\!] = [\![F_1]\!] \otimes [\![F_2]\!]$$

$$\text{where } M_1 \otimes M_2 = \{m_1 \oplus m_2 \mid m_1 \in M_1, m_2 \in M_2\}$$

Table 1: Denotational semantics

Given a BPP net $N = (S, A, T)$, the semantics of F is a set of markings on S. Formally, let $[\![-]\!] : \mathscr{F}_A \to \wp(\mathscr{M}(S))$ be the denotational semantics function, defined in Table 1.

The semantics of tt is $\mathscr{M}(S)$: every marking belongs to $[\![tt]\!]$. The semantics of ff is $\emptyset$: no marking belongs to $[\![ff]\!]$. The logical operator of conjunction $\_ \wedge \_$ is interpreted as intersection $\_ \cap \_$, while, symmetrically, disjunction $\_ \vee \_$ is interpreted as set union $\_ \cup \_$. The semantics of $\neg F$ is the set of all the markings that do not belong to $[\![F]\!]$, i.e., the complement of $[\![F]\!]$ w.r.t. the universe $\mathscr{M}(S)$.

The semantics of $\langle a \rangle F$ is the set of all the places that can perform a and, in doing so, reach a marking m such that $d(m) \in [\![F]\!]$, where $d(m)$ is the marking obtained from m by removing all the deadlock places. For instance, the semantics of the formula $\langle a \rangle tt$ is the set of places able to perform a. The semantics of $[a]F$ is the set of all the places (always including also the empty marking θ) that, by performing a, can only reach markings satisfying F. Note that a place s, which is unable to perform a altogether, is in the semantics of $[a]F$, for any F, because the universal quantification in the definition of its semantic is vacuously satisfied; for this reason, the empty marking is in the semantics of $[a]F$ for any F. For instance, the semantics of the formula $[a]ff$ is the set (including θ) of all the places that cannot perform a. The semantics of $F_1 \otimes F_2$ is the set of markings of the form $m_1 \oplus m_2$ such that $m_1 \in [\![F_1]\!]$ and $m_2 \in [\![F_2]\!]$.

Definition 24. (HTML satisfaction relation) Given a BPP net $N = (S, A, T)$, we say that a marking $m \in \mathscr{M}(S)$ satisfies formula F, written $m \models F$, if $d(m) \in [\![F]\!]$. □

Of course, this means that m and $d(m)$ satisfy the same HTML formulae, because $d(d(m)) = d(m)$. In particular, note that a deadlock place s satisfies the same formulae as θ because $d(s) = \theta$.

Example 13. Let us consider the BPP net in Figure 4 and the formula $F_1 \otimes F_2$, where $F_1 = \langle a \rangle [b] \langle c \rangle tt$ and $F_2 = [\{a, c\}]ff$. The marking $s_1 \oplus s_2$ is in the semantics of $F_1 \otimes F_2$ because $s_1 \in [\![F_1]\!]$ and $s_2 \in [\![F_2]\!]$. Indeed, the semantics of $F_1 \otimes F_2$ is given by $\{s_1, s_4\} \otimes \{\theta, s_2, s_5, s_6, s_7\} = \{s_1, s_1 \oplus s_2, s_1 \oplus s_5, s_1 \oplus s_6, s_1 \oplus s_7, s_4, s_4 \oplus s_2, s_4 \oplus s_5, s_4 \oplus s_6, s_4 \oplus s_7\}$: these are all the markings *satisfying* $F_1 \otimes F_2$, because there are no deadlock places. □

Example 14. Let us consider the BPP net in Figure 3(c) and the formula $F = \langle a \rangle tt$. The semantics of $F \otimes F$ is given by $\{s_6, s_8\} \otimes \{s_6, s_8\} = \{2 \cdot s_6, s_6 \oplus s_8, 2 \cdot s_8\}$. However the set of markings satisfying $F \otimes F$ is infinite: for instance, any marking of the form $s_6 \oplus s_8 \oplus k \cdot s_7$ satisfies that formula, for each $k \in \mathbb{N}$. □

We are now ready to prove the coherence theorem: two markings are h-team bisimilar if and only if they satisfy the same HTML formulae.

Proposition 25. Let $N = (S, A, T)$ be a BPP net. If $m_1 \sim_h^\oplus m_2$, then m_1 and m_2 satisfy the same HTML formulae, i.e., $\{F_1 \in \mathscr{F}_A \mid m_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid m_2 \models F_2\}$.

PROOF. Let us assume that $m_1 \sim_h^\oplus m_2$. We will prove that, for any $F \in \mathscr{F}_A$, if $m_1 \models F$ then also $m_2 \models F$. This is enough because, as $\sim_h^\oplus$ is symmetric, this justifies that if $m_2 \models F$ then also $m_1 \models F$, and so m_1 and m_2 satisfy the same HTML formulae.

The proof is by induction on the structure of F, where the first two cases are the base cases of the induction.

- $F = tt$: if $m_1 \models tt$, then $d(m_1) \in [\![tt]\!]$. Since also $d(m_2) \in [\![tt]\!]$, we get $m_2 \models tt$, too.
- $F = ff$: since no marking satisfies false, $m_1 \not\models ff$ and also $m_2 \not\models ff$.
- $F = F_1 \wedge F_2$: since $m_1 \models F_1 \wedge F_2$, it follows that $m_1 \models F_1$ and $m_1 \models F_2$; by induction, we can assume that also $m_2 \models F_1$ and $m_2 \models F_2$; hence, also $m_2 \models F_1 \wedge F_2$, as required.
- $F = F_1 \vee F_2$: similar to the above.
- $F = \langle a \rangle G$: if $m_1 \models \langle a \rangle G$, then $d(m_1) = s_1$ such that $s_1 \in [\![\langle a \rangle G]\!]$; hence, there exists a marking m_1' such that $s_1 \xrightarrow{a} m_1'$ and $m_1' \models G$. As $m_1 \sim_h^\oplus m_2$, it follows that $d(m_2) = s_2$ such that $s_1 \sim_h s_2$. Therefore, by definition of $\sim_h$, there exists a marking m_2' such that $s_2 \xrightarrow{a} m_2'$ and $m_1' \sim_h^\oplus m_2'$. Since $m_1' \sim_h^\oplus m_2'$ and $m_1' \models G$, we can apply induction (because G is a subformula) and conclude that also $m_2' \models G$; hence, $s_2 \in [\![\langle a \rangle G]\!]$, and so $m_2 \models \langle a \rangle G$, as required.
- $F = [a]G$: if $m_1 \models [a]G$, then $d(m_1) = s_1$ such that $s_1 \in [\![[a]G]\!]$; hence, for all m_1' such that $s_1 \xrightarrow{a} m_1'$, it follows that $m_1' \models G$. As $m_1 \sim_h^\oplus m_2$, it follows that $d(m_2) = s_2$ such that $s_1 \sim_h s_2$. Since $s_1 \sim_h s_2$, for each m_2' such that $s_2 \xrightarrow{a} m_2'$, there exists m_1' such that $s_1 \xrightarrow{a} m_1'$ such that $m_1' \sim_h^\oplus m_2'$. Now, since $m_1' \sim_h^\oplus m_2'$ and $m_1' \models G$, by induction, it follows also that $m_2' \models G$. Hence, for all m_2' such that $s_2 \xrightarrow{a} m_2'$, we have that $m_2' \models G$; therefore, $s_2 \in [\![[a]G]\!]$, and so $m_2 \models [a]G$, as required.
- $F = \neg F'$: since $m_1 \models \neg F'$, it follows that $m_1 \not\models F'$. By induction, as F' is a subformula, if m_1 does not satisfy F', then also m_2 does not satisfy F', and so $m_2 \models \neg F'$, as required.
- $F = F_1 \otimes F_2$: $m_1 \models F_1 \otimes F_2$ only if there exists m_1' and m_1'' such that $m_1 = m_1' \oplus m_1''$, $m_1' \models F_1$ and $m_1'' \models F_2$. As $m_1 \sim_h^\oplus m_2$, there exists m_2' and m_2'' such that $m_2 = m_2' \oplus m_2''$ and $m_1' \sim_h^\oplus m_2'$ and $m_1'' \sim_h^\oplus m_2''$. By induction, $m_2' \models F_1$ and $m_2'' \models F_2$; therefore, also $m_2 \models F_1 \otimes F_2$, as required.

□

Lemma 1. Let $N = (S, A, T)$ be a BPP net. Let $r_1, r_2 \in S \cup \{\theta\}$. If r_1 and r_2 satisfy the same HTML formulae, i.e., $\{F_1 \in \mathscr{F}_A \mid r_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid r_2 \models F_2\}$, then $r_1 \sim_h r_2$.

PROOF. We want to prove that $R = \{(r, r') \mid r, r' \in S \cup \{\theta\}, r$ and r' satisfy the same HTML formulae$\}$ is an h-team bisimulation, hence proving that two places (or one place and the empty marking) that satisfy the same formulae are h-team bisimilar.

Assume $(r_1, r_2) \in R$ and $r_1 \xrightarrow{a} m_1$. We will prove that there exists some m_2 such that $r_2 \xrightarrow{a} m_2$ and $(m_1, m_2) \in R^{\oplus}$. Since R is symmetric, this is enough for proving that R is an h-team bisimulation.

Assume, towards a contradiction, that there exists no m_2 such that $r_2 \xrightarrow{a} m_2$ and $(m_1, m_2) \in R^{\oplus}$. Since the net is finite, the set $\{m \in \mathscr{M}(S) \mid r_2 \xrightarrow{a} m\}$ is finite; let us denote such a set with $\{m'_1, m'_2, \ldots, m'_k\}$, with $k \geq 0$. By assumption, for $j = 1, \ldots, k$, none of the m'_j is such that $(m_1, m'_j) \in R^{\oplus}$. Therefore, by looking at Algorithm 1 (which is applicable as R is an equivalence relation), one of the following two cases is possible:

(a) there is a place p_j in the residual of $d(m_1)$ that has no R-match in the residual of $d(m'_j)$; or, vice versa,

(b) there is a p_j in the residual of $d(m'_j)$ that has no R-match in the residual of $d(m_1)$.

In case (a), assume that $dom(m'_j)$ has $h_j \geq 1$ non-deadlock places which are not R-related to p_j, namely $\{s^j_1, \ldots, s^j_{h_j}\} \subseteq dom(m'_j)$. Hence, for each $s^j_i \in m'_j$, for $i = 1, \ldots, h_j$, there is an HTML formula F^j_i such that $p_j \models F^j_i$ and $s^j_i \not\models F^j_i$. Let m' be the marking composed of all the elements s in m_1 such that $(s, p_j) \in R$; to be precise, any $s \in m'$ is such that $(s, p_j) \in R$, and any $s \in m_1 \ominus m'$ is such that $(s, p_j) \notin R$. Then,

$$m' \models G^{l_j}_j = \underbrace{G_j \otimes \ldots \otimes G_j}_{l_j \text{ times}},$$

where $G_j = F^j_1 \wedge \ldots \wedge F^j_{h_j}$ and $l_j = |m'|$. By Definition 24, also $m_1 \models G^{l_j}_j \otimes nn^{n-l_j}$, where $n = |d(m_1)|$ and nn^{n-l_j} is the shorthand for a formula of the form $\langle b_1 \rangle tt \otimes \ldots \otimes \langle b_{n-l_j} \rangle tt$, for suitably chosen $b_1, \ldots, b_{n-l_j}$.[2] On the contrary, $m'_j \not\models G^{l_j}_j \otimes nn^{n-l_j}$ because in m'_j there are less than l_j elements which are R-related to p_j and any other s^j_i is such that $s^j_i \not\models F^j_i$ and so $s^j_i \not\models G_j$.

In case (b), assume that $dom(m_1)$ has $h \geq 1$ non-deadlock places which are not R-related to $p_j \in m'_j$, namely $\{s^1_1, \ldots, s^1_h\} \subseteq dom(m_1)$. Hence, for each $s^1_i \in m_1$, for $i = 1, \ldots, h$, there is an HTML formula F^j_i such that $p_j \models F^j_i$ and $s^1_i \not\models F^j_i$. Let m' be the marking composed of all the elements s in m'_j such that $(s, p_j) \in R$; to be precise, any $s \in m'$ is such that $(s, p_j) \in R$, and any $s \in m'_j \ominus m'$ is such that $(s, p_j) \notin R$. Then,

[2] Since each of the $n - l_j$ places in $d(m_1) \ominus m'$ is not a deadlock, each one of them satisfies some formula of the form $\langle b \rangle tt$ for some suitably chosen action b.

$$m' \models H_j^{l_j} = \underbrace{H_j \otimes \ldots \otimes H_j}_{l_j \text{ times}},$$

where $H_j = F_1^j \wedge \ldots \wedge F_h^j$ and $l_j = |m'|$. By Definition 24, also $m'_j \models H_j^{l_j} \otimes nn^{n_j - l_j}$, where $n_j = |d(m'_j)|$. On the contrary, $m_1 \not\models H_j^{l_j} \otimes nn^{n_j - l_j}$ because in m_1 there are less than l_j elements which are R-related to p_j and any other s_i^1 is such that $s_i^1 \not\models F_i^j$ and so $s_i^1 \not\models H_j$.

Finally, take the formula $G = \langle a \rangle (K_1 \wedge K_2 \wedge \ldots \wedge K_k)$, where, for $j = 1, \ldots, k$,

- the formula K_j is $G_j^{l_j} \otimes nn^{n-l_j}$, if case (a) applies; or
- the formula K_j is $\neg(H_j^{l_j} \otimes nn^{n_j - l_j})$, if case (b) applies.

It is easy to see that $r_1 \models G$, because $m_1 \models K_j$ for $j = 1, \ldots, k$; on the contrary, $r_2 \not\models G$, because, for $j = 1, \ldots, k$, $m'_j \not\models K_j$, hence contradicting the previous assumption that r_1 and r_2 satisfy the same formulae. (In case $k = 0$, $G = \langle a \rangle tt$.) □

Proposition 26. Let $N = (S, A, T)$ be a BPP net. If m_1 and m_2 satisfy the same HTML formulae, i.e., $\{F_1 \in \mathscr{F}_A \mid m_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid m_2 \models F_2\}$, then $m_1 \sim_h^{\oplus} m_2$.

PROOF. We actually prove the contranominal: if two markings are not related by $\sim_h^{\oplus}$, then they do not satisfy the same HTML formulae. Two markings m_1 and m_2 are not h-team bisimilar if, after removing all the deadlock places occurring in m_1 and m_2, the resulting markings $d(m_1)$ and $d(m_2)$ are such that there is no h-team bisimulation-preserving bijection among the elements of these two markings. This may happen because either $d(m_1)$ and $d(m_2)$ have not the same size or, w.l.o.g., there is an element s'_i in (the residual of) $d(m_1)$ that has not h-bisimilar match in (the residual of) $d(m_2)$.

In the former case, assume that $|d(m_1)| = n \neq h = |d(m_2)|$ for some $n \geq 1$. Let us assume that $d(m_1) = s'_1 \oplus \ldots s'_n$; since s'_i is not a deadlock, $s'_i \models \langle a_i \rangle tt$ for some action $a_i \in A$. Then, the HTML formula

$$\langle a_1 \rangle tt \otimes \ldots \otimes \langle a_n \rangle tt$$

is such that $m_1 \models \langle a_1 \rangle tt \otimes \ldots \otimes \langle a_n \rangle tt$, while $m_2 \not\models \langle a_1 \rangle tt \otimes \ldots \otimes \langle a_n \rangle tt$ because m_2 has $h \neq n$ non-deadlock places, and so m_1 and m_2 do not satisfy the same HTML formulae.

In the latter case, let s be the element of the residual of $d(m_1)$ that has no h-team bisimilar match in the residual of $d(m_2)$. Assume that $dom(d(m_2))$ has $k \geq 1$ places which are not h-team bisimilar to s, namely $\{s'_1, \ldots, s'_k\} \subseteq dom(d(m_2))$. Hence, by (the contranominal of) Lemma 1, for each $s'_j \in d(m_2)$, there is an HTML formula F_j such that $s \models F_j$ and $s'_j \not\models F_j$, for $j = 1, \ldots, k$. Let m'_1 be the marking composed of all the elements s' in $d(m_1)$ such that $s' \sim_h s$; to be precise, any $s' \in m'_1$ is such that $s' \sim_h s$, and any $s' \in d(m_1) \ominus m'_1$ is such that $s' \nsim_h s$. Then,

$$m'_1 \models G^h = \underbrace{G \otimes \ldots \otimes G}_{h \text{ times}},$$

where $G = F_1 \wedge \ldots \wedge F_k$ and $h = |m'_1|$. By Definition 24, also $d(m_1) \models G^h \otimes nn^l$, where $l = |d(m_1)| - |m'_1|$ and nn^l is the shorthand for a formula of the form $\langle b_1 \rangle tt \otimes \ldots \otimes \langle b_l \rangle tt$, for suitably chosen $b_1, \ldots, b_l$. On the contrary, $d(m_2) \not\models G^h \otimes nn^l$ because in $d(m_2)$ there are less than h elements which are h-team bisimilar to s and any other s'_j is such that $s'_j \not\models F_j$ and so $s'_j \not\models G$. In conclusion, since m_1 satisfies the same formulae as $d(m_1)$ and

$dec(\mathbf{0})$	$=$	θ	$dec(\mu.p)$	$=$	$\{\mu.p\}$
$dec(p+p')$	$=$	$\{p+p'\}$	$dec(C)$	$=$	$\{C\}$
$dec(p\,\|\,p')$	$=$	$dec(p) \oplus dec(p')$			

Table 2: Decomposition function

m_2 satisfies the same formulae as $d(m_2)$, we get that m_1 and m_2 do not satisfy the same HTML formulae. □

Theorem 6. (Coherence) Let $N = (S, A, T)$ be a BPP net. It holds that $m_1 \sim_h^{\oplus} m_2$ if and only if $\{F_1 \in \mathscr{F}_A \mid m_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid m_2 \models F_2\}$.

PROOF. Direct consequence of Propositions 25 and 26. □

6. Axiomatizing H-Team Bisimilarity

Now we want to show that h-team bisimilarity can be axiomatized. This can be done because BPP nets can be "alphabetized" by means of the process algebra BPP (where BPP is the acronym of *Basic Parallel Processes*) and was originally studied in [12]. BPP is a simple CCS [43, 27] subcalculus (without the restriction operator) whose processes cannot communicate. We actually study the variant BPP in [28] which requires guarded summation (as in SBPP [17, 20] or BPP_g [12]) and also guarded recursion.

6.1. The BPP Process Algebra

Its syntax is defined as follows. Let *Act* be a finite set of *actions*, ranged over by μ, and let $\mathscr{C}$ be a finite set of *constants*, disjoint from *Act*, ranged over by $A, B, C, \ldots$. The size of the sets *Act* and $\mathscr{C}$ is not important: we assume that they can be chosen are as large as needed. The BPP *terms* are generated from actions and constants by the following abstract syntax (using three syntactic categories):

$$\begin{array}{lllllll} s & ::= & \mathbf{0} & \mid & \mu.p & \mid & s+s \qquad \textit{guarded processes} \\ q & ::= & s & \mid & C & & \qquad \textit{sequential processes} \\ p & ::= & q & \mid & p\,|\,p & & \qquad \textit{parallel processes} \end{array}$$

where $\mathbf{0}$ is the empty process, $\mu.p$ is a process where action $\mu \in Act$ prefixes the residual p ($\mu.-$ is the *action prefixing* operator), $s_1 + s_2$ denotes the alternative composition of s_1 and s_2 ($- + -$ is the *choice* operator), $p_1 \,|\, p_2$ denotes the asynchronous parallel composition of p_1 and p_2 and C is a constant. A constant C may be equipped with a definition, but this must be a guarded process, i.e., in the syntactic category s: $C \doteq s$. A term p is a BPP *process* if each constant in $Const(p)$ (the set of constants used by p; see [28] for details) is equipped with a defining equation (in syntactic category s). The set of BPP processes is denoted by $\mathscr{P}_{BPP}$, the set of its sequential processes, i.e., of the processes in syntactic category q, by $\mathscr{P}_{BPP}^{seq}$, while the set of its guarded processes, i.e., of the processes in syntactic category s, by $\mathscr{P}_{BPP}^{grd}$,

$$
\begin{array}{rcll}
[\![\mathbf{0}]\!]_I & = & (\emptyset,\emptyset,\emptyset,\theta) & \\
[\![\mu.p]\!]_I & = & (S,A,T,\{\mu.p\}) & \text{given } [\![p]\!]_I = (S',A',T',dec(p)) \text{ and} \\
 & & & S = \{\mu.p\} \cup S',\ A = \{\mu\} \cup A', \\
 & & & T = \{(\{\mu.p\},\mu,dec(p))\} \cup T' \\
[\![p_1 + p_2]\!]_I & = & (S,A,T,\{p_1+p_2\}) & \text{given } [\![p_i]\!]_I = (S_i,A_i,T_i,dec(p_i)) \text{ for } i = 1,2, \text{ and} \\
 & & & S = \{p_1+p_2\} \cup S'_1 \cup S'_2, \text{ with, for } i = 1,2, \\
 & & & S'_i = \begin{cases} S_i & \exists t \in T_i \text{ such that } t^\bullet(p_i) > 0 \\ S_i \setminus \{p_i\} & \text{otherwise} \end{cases} \\
 & & & A = A_1 \cup A_2,\ T = T' \cup T'_1 \cup T'_2, \text{ with, for } i = 1,2, \\
 & & & T'_i = \begin{cases} T_i & \exists t \in T_i\,.\,t^\bullet(p_i) > 0 \\ T_i \setminus \{t \in T_i \mid {}^\bullet t(p_i) > 0\} & \text{otherwise} \end{cases} \\
 & & & T' = \{(\{p_1+p_2\},\mu,m) \mid (\{p_i\},\mu,m) \in T_i, i = 1,2\} \\
[\![C]\!]_I & = & (\{C\},\emptyset,\emptyset,\{C\}) & \text{if } C \in I \\
[\![C]\!]_I & = & (S,A,T,\{C\}) & \text{if } C \notin I, \text{ given } C \doteq p \text{ and } [\![p]\!]_{I\cup\{C\}} = (S',A',T',dec(p)) \\
 & & & A = A', S = \{C\} \cup S'', \text{ where} \\
 & & & S'' = \begin{cases} S' & \exists t \in T'\,.\,t^\bullet(p) > 0 \\ S' \setminus \{p\} & \text{otherwise} \end{cases} \\
 & & & T = \{(\{C\},\mu,m) \mid (\{p\},\mu,m) \in T'\} \cup T'' \text{ where} \\
 & & & T'' = \begin{cases} T' & \exists t \in T'\,.\,t^\bullet(p) > 0 \\ T' \setminus \{t \in T' \mid {}^\bullet t(p) > 0\} & \text{otherwise} \end{cases} \\
[\![p_1 \mid p_2]\!]_I & = & (S,A,T,m_0) & \text{given } [\![p_i]\!]_I = (S_i,A_i,T_i,m_i) \text{ for } i = 1,2, \text{ and where} \\
 & & & S = S_1 \cup S_2,\ A = A_1 \cup A_2,\ T = T_1 \cup T_2,\ m_0 = m_1 \oplus m_2
\end{array}
$$

Table 3: Denotational net semantics

The net semantics for the process algebra BPP, originally outlined in [28], is such that the set of places S_{BPP} is the set of the sequential BPP processes, without $\mathbf{0}$, i.e., $S_{BPP} = \mathscr{P}_{BPP}^{seq} \setminus \{\mathbf{0}\}$. The decomposition function $dec : \mathscr{P}_{BPP} \to \mathscr{M}(S_{BPP})$, mapping process terms to markings, is defined in Table 2. An easy induction proves that for any $p \in \mathscr{P}_{BPP}$, $dec(p)$ is a finite multiset of sequential processes. Note that, if $C \doteq \mathbf{0}$, then $\theta = dec(\mathbf{0}) \neq dec(C) = \{C\}$; moreover, note that $\theta = dec(\mathbf{0}) \neq dec(\mathbf{0}+\mathbf{0}) = \{\mathbf{0}+\mathbf{0}\}$, which is a deadlock place.

Now we provide a construction of the net system $[\![p]\!]_\emptyset$ associated with process p, which is compositional and denotational in style. The details of the construction are outlined in Table 3. The mapping is parametrized by a set of constants that have already been found while scanning p; such a set is initially empty and it is used to avoid looping on recursive constants. The definition is syntax driven and also the places of the constructed net are syntactic objects, i.e., BPP sequential process terms. For instance, the net system $[\![a.\mathbf{0}]\!]_\emptyset$ is a net composed of one single marked place, namely process $a.\mathbf{0}$, and one single transition $(\{a.\mathbf{0}\},a,\theta)$. A bit of care is needed in the rule for choice: in order to include only strictly necessary places and transitions, the initial place p_1 (or

p_2) of the subnet $[\![p_1]\!]_I$ (or $[\![p_2]\!]_I$) is to be kept in the net for $p_1 + p_2$ only if there exists a transition reaching place p_1 (or p_2) in $[\![p_1]\!]_I$ (or $[\![p_2]\!]_I$), otherwise p_1 (or p_2) can be safely removed in the new net. Similarly, for the rule for constants.

Example 15. Consider the BPP process *SC* for a semi-counter, whose definition is
$SC \doteq inc.(SC \,|\, dec.\mathbf{0})$.
We have that
$[\![SC]\!]_{\{SC\}} = (\{SC\}, \emptyset, \emptyset, \{SC\})$, and
$[\![dec.\mathbf{0}]\!]_{\{SC\}} = (\{dec.\mathbf{0}\}, \{dec\}, \{(\{dec.\mathbf{0}\}, dec, \theta)\}, \{dec.\mathbf{0}\})$.
Therefore, the net $[\![SC \,|\, dec.\mathbf{0}]\!]_{\{SC\}}$ is
$(\{SC, dec.\mathbf{0}\}, \{dec\}, \{(\{dec.\mathbf{0}\}, dec, \theta)\}, \{SC, dec.\mathbf{0}\})$.
The net $[\![inc.(SC \,|\, dec.\mathbf{0})]\!]_{\{SC\}}$ is
$(\{inc.(SC \,|\, dec.\mathbf{0}), SC, dec.\mathbf{0}\}, \{inc, dec\}, \{(\{inc.(SC \,|\, dec.\mathbf{0})\}, inc, \{SC, dec.\mathbf{0}\}),$
$(\{dec.\mathbf{0}\}, dec, \theta)\}, \{inc.(SC \,|\, dec.\mathbf{0})\})$.
Finally, the net $[\![SC]\!]_\emptyset$ is
$(\{SC, dec.\mathbf{0}\}, \{inc, dec\}, \{(\{SC\}, inc, \{SC, dec.\mathbf{0}\}), (\{dec.\mathbf{0}\}, dec, \theta)\}, \{SC\})$,
which is (isomorphic to) the net in Figure 1(a), where s_1 is *SC* and s_2 is $dec.\mathbf{0}$. □

The net semantics for BPP is such that:

- the semantics of a BPP process term p is a BPP net system $[\![p]\!]_\emptyset$, whose initial marking is $dec(p)$; moreover,
- for any BPP net system $N(m_0)$ (which is dynamically reduced), there exists a BPP process term $p_{N(m_0)}$ such that its semantics $[\![p_{N(m_0)}]\!]_\emptyset$ is a net isomorphic to $N(m_0)$ (*Representability Theorem*).

Therefore, thanks to these results (proved in [28]), we can conclude that the BPP process algebra truly represents the class of BPP nets. Hence, we can transpose the definition of h-team bisimilarity from BPP nets to BPP process terms in a simple way.

Definition 25. Two BPP processes p and q are h-team bisimilar, denoted $p \sim_h^\oplus q$, if, by taking the (union of the) nets $[\![p]\!]_\emptyset$ and $[\![q]\!]_\emptyset$, we have that $dec(p) \sim_h^\oplus dec(q)$. □

Of course, for sequential BPP processes, h-team bisimulation equivalence $\sim_h^\oplus$ coincides with h-team bisimilarity on places $\sim_h$.

Thanks to Definition 25, we can now perform the usual process algebraic study of a behavioral equivalence: to prove that it is a congruence for the operators of the BPP process algebra, to study its algebraic properties and, finally, to define a (possibly finite) sound and complete, axiomatization for it. These will be the subject of the next subsections.

6.2. Congruence

Now we show that team equivalence is a congruence for all the BPP operators.

Proposition 27. (Congruence)
1) For each $p, q, r \in \mathscr{P}_{BPP}^{grd}$, if $p \sim_h q$, then $p + r \sim_h q + r$.
2) For each $p, q \in \mathscr{P}_{BPP}$, if $p \sim_h^\oplus q$, then $\mu.p \sim_h \mu.q$ for all $\mu \in Act$.
3) For every $p, q, r \in \mathscr{P}_{BPP}$, if $p \sim_h^\oplus q$, then $p \,|\, r \sim_h^\oplus q \,|\, r$.

PROOF. 1) Assume R is an h-team bisimulation such that $(p,q) \in R$.[3] It is very easy to check that, for each $r \in \mathscr{P}_{BPP}^{grd}$, the relation $R_r = \{(p+r, q+r)\} \cup R \cup \mathscr{I}_r$ is an h-team bisimulation, where $\mathscr{I}_r = \{(r', r') \mid r' \in reach(dec(r))\}$.

2) Assume R is an h-team bisimulation such that $(dec(p), dec(q)) \in R^{\oplus}$. Consider, for each $\mu \in Act$, relation $R_\mu = \{(\mu.p, \mu.q)\} \cup R$. It is very easy to check that R_μ is an h-team bisimulation on places.

3) By induction on the size of $dec(p)$. If $|dec(p)| = 0$, then $dec(p) = \theta$; as $p \sim_h^{\oplus} q$, it follows that $\theta \sim_h^{\oplus} dec(q)$. Hence, the thesis follows trivially:
$dec(p\,|\,r) = \theta \oplus dec(r) \sim_h^{\oplus} dec(q) \oplus dec(r) = dec(q\,|\,r)$.

If $|dec(p)| = k+1$ for some $k \geq 0$, then there exist p_1 and p_2 such that $dec(p) = p_1 \oplus dec(p_2)$. If $p_1 \sim_h \theta$, then $dec(p) \sim_h^{\oplus} dec(p_2)$; as $|dec(p_2)| = k$ and $p_2 \sim_h^{\oplus} p \sim_h^{\oplus} q$, by induction, we have that $p_2\,|\,r \sim^{\oplus} q\,|\,r$. Then, the thesis follows trivially:
$dec(p\,|\,r) = p_1 \oplus dec(p_2) \oplus dec(r) \sim_h^{\oplus} dec(p_2) \oplus dec(r) = dec(p_2\,|\,r) \sim_h^{\oplus} dec(q\,|\,r)$.
Otherwise, if $p_1 \nsim_h \theta$, by Definition 22, there exist q_1, q_2 such that $dec(q) = q_1 \oplus dec(q_2)$, $p_1 \sim_h q_1$ and $dec(p_2) \sim_h^{\oplus} dec(q_2)$. Since $|dec(p_2)| = k$ and $p_2 \sim_h^{\oplus} q_2$, by induction, we have that $p_2\,|\,r \sim^{\oplus} q_2\,|\,r$. As $p_1 \sim_h q_1$, by Definition 22, we have
$dec(p\,|\,r) = p_1 \oplus dec(p_2\,|\,r) \sim_h^{\oplus} q_1 \oplus dec(q_2\,|\,r) = dec(q\,|\,r)$. So, $p\,|\,r \sim_h^{\oplus} q\,|\,r$. □

Still there is one construct missing: recursion, defined over guarded terms only. Consider an extension of BPP where terms can be constructed using variables, such as $x, y, \ldots$: this defines an "open" BPP. We use the notation $p(x_1, \ldots, x_n)$ to state that term p is open on the tuple of variables $(x_1, \ldots, x_n)$. For instance, $p_1(x) = a.(b.\mathbf{0} + c.x) + d.x$ and $p_2(x) = d.x + a.(c.x + b.\mathbf{0})$ are open guarded BPP terms.

Definition 26. (Open BPP) Let $Var = \{x, y, z, \ldots\}$ be a finite set of variables. The BPP *open terms* are generated from actions, constants and variables by the following abstract syntax (using three syntactic categories):

$$\begin{array}{llllllll} s & ::= & \mathbf{0} & \mid & \mu.p & \mid & s+s & \textit{guarded open processes} \\ q & ::= & s & \mid & C & \mid & x & \textit{sequential open processes} \\ p & ::= & q & \mid & p\,|\,p & & & \textit{parallel open processes} \end{array}$$

where x is any variable taken from Var. The *open* net semantics for open BPP extends the net semantics in Table 3 with $[\![x]\!]_I = (\{x\}, \emptyset, \emptyset, \{x\})$, so that, e.g., the semantics of $a.x$ is the net $(\{a.x, x\}, \{a\}, \{(a.x, a, x)\}, a.x)$. □

H-team bisimulation equivalence can be extended to open terms as follows. An open term $p(x_1, \ldots, x_n)$ can be *closed* by means of a substitution:

$$p(x_1, \ldots, x_n)\{r_1/x_1, \ldots, r_n/x_n\}$$

with the effect that each occurrence of the variable x_i (within p and the body of each constant in $Const(p)$) is replaced by the *closed* BPP sequential process r_i, for $i = 1, \ldots, n$. For instance, $p_1(x)\{d.\mathbf{0}/x\} = a.(b.\mathbf{0} + c.d.\mathbf{0}) + d.d.\mathbf{0}$.

[3] The case when p or q is $\mathbf{0}$ is trivial; in such as case, R is either $\{(\theta, \theta)\}$ (if both p and q are $\mathbf{0}$) or $\{(\theta, q)\}$ or $\{(p, \theta)\}$.

A natural extension of h-team bisimilarity $\sim_h$ over open *guarded* terms is as follows: $p(x_1,\ldots,x_n) \sim_h q(x_1,\ldots,x_n)$ if for all tuples of (closed) BPP sequential terms $(r_1,\ldots,r_n)$, $p(x_1,\ldots,x_n)\{r_1/x_1,\ldots,r_n/x_n\} \sim_h q(x_1,\ldots,x_n)\{r_1/x_1,\ldots,r_n/x_n\}$. E.g., it is easy to see that $p_1(x) \sim_h p_2(x)$. As a matter of fact, for all r,

$p_1(x)\{r/x\} = a.(b.\mathbf{0}+c.r)+d.r \sim_h d.r+a.(c.r+b.\mathbf{0}) = p_2(x)\{r/x\}$,

which can be easily proved by means of the algebraic properties (discussed in the next subsection) and the congruence ones of $\sim_h$.

For simplicity's sake, let us now restrict our attention to open guarded terms using a single undefined variable. We can *recursively close* an open term $p(x)$ by means of a recursively defined constant. For instance, $A \doteq p(x)\{A/x\}$. The resulting process constant A is a closed BPP sequential process. By saying that h-team bisimilarity is a congruence for recursion we mean what is stated in the following theorem.

Theorem 7. Let p and q be two open guarded BPP terms, with one variable x at most. Let $A \doteq p\{A/x\}$, $B \doteq q\{B/x\}$ and $p \sim_h q$. Then $A \sim_h B$.

PROOF. Consider $R = \{(r\{A/x\}, r\{B/x\}) \mid r \in reach(dec(p)) \cup reach(dec(q))\}$. Note that when r is x, we get $(A,B) \in R$. The proof that R is an h-team bisimulation up to $\sim_h$ [30] is not difficult. By symmetry, it is enough to prove that if $r\{A/x\} \xrightarrow{\mu} m_1$, then $r\{B/x\} \xrightarrow{\mu} m_2$ such that $m_1 (\sim_h R \sim_h)^{\oplus} m_2$. The proof proceeds by induction on the definition of the net for $r\{A/x\}$.

- $r = \mu.r'$. In this case, $r\{A/x\} = \mu.r'\{A/x\} \xrightarrow{\mu} dec(r')\{A/x\}$. Similarly, $r\{B/x\} = \mu.r'\{B/x\} \xrightarrow{\mu} dec(r')\{B/x\}$, and $(dec(r')\{A/x\}, dec(r')\{B/x\}) \in R^{\oplus}$.

- $r = D$, with $D \doteq s$. So, $r\{A/x\} \doteq s\{A/x\}$ and $r\{B/x\} \doteq s\{B/x\}$. If $r\{A/x\} \xrightarrow{\mu} m_1$, then this is possible only if $s\{A/x\} \xrightarrow{\mu} m_1$. Since s is guarded, $s\{A/x\} \xrightarrow{\mu} m_1$ is possible only if $s \xrightarrow{\mu} m$ with $m_1 = m\{A/x\}$. Therefore, also $s\{B/x\} \xrightarrow{\mu} m\{B/x\}$ is derivable, and also $r\{B/x\} \xrightarrow{\mu} m\{B/x\}$, with $(m\{A/x\}, m\{B/x\}) \in R^{\oplus}$.

- $r = r_1 + r_2$. In this case, $r\{A/x\} = r_1\{A/x\} + r_2\{A/x\}$. A transition from $r\{A/x\}$, e.g., $r_1\{A/x\} + r_2\{A/x\} \xrightarrow{\mu} m_1$, is derivable only if $r_i\{A/x\} \xrightarrow{\mu} m_1$ for some $i = 1,2$. Without loss of generality, assume the transition is due to $r_1\{A/x\} \xrightarrow{\mu} m_1$. Since r_1 is guarded, transition $r_1\{A/x\} \xrightarrow{\mu} m_1$ is derivable because $r_1 \xrightarrow{\mu} m$, with $m_1 = m\{A/x\}$. Therefore, also $r_1\{B/x\} \xrightarrow{\mu} m\{B/x\}$ is derivable, as well $r\{B/x\} = r_1\{B/x\} + r_2\{B/x\} \xrightarrow{\mu} m\{B/x\}$, with $(m\{A/x\}, m\{B/x\}) \in R^{\oplus}$.

- $r = x$. We have $r\{A/x\} = A$ and $r\{B/x\} = B$. We prove that for each $A \xrightarrow{\mu} m_1$, there exists m_2 such that $B \xrightarrow{\mu} m_2$ with $m_1 (\sim_h R \sim_h)^{\oplus} m_2$. By hypothesis, $A \doteq p\{A/x\}$, hence also $p\{A/x\} \xrightarrow{\mu} m_1$ is a transition in the net for $p\{A/x\}$; since p is guarded, $p\{A/x\} \xrightarrow{\mu} m_1$ is possible only if $p \xrightarrow{\mu} m$ with $m_1 = m\{A/x\}$. Therefore, also $p\{B/x\} \xrightarrow{\mu} m\{B/x\}$ is derivable. But we also have that $p \sim_h q$, so $p \xrightarrow{\mu} m$ can be matched by $q \xrightarrow{\mu} m'$ with $m \sim_h^{\oplus} m'$. Hence, $q\{B/x\} \xrightarrow{\mu} m'\{B/x\}$

is derivable with $m\{B/x\} \sim_h^{\oplus} m'\{B/x\}$. Since $B \doteq q\{B/x\}$, also $B \xrightarrow{\mu} m'\{B/x\}$ is a transition with $m_1 \sim_h^{\oplus} m\{A/x\} R^{\oplus} m\{B/x\} \sim_h^{\oplus} m'\{B/x\}$, as required. □

The extension to the case of open terms with multiple undefined constants, e.g., $p(x_1, \ldots, x_n)$ can be obtained in a standard way [43, 27].

6.3. Algebraic Laws

Now we list the algebraic properties of h-team bisimulation equivalence. On sequential processes we have the following algebraic laws.

Proposition 28. (Laws of the choice op.) For each $p, q, r \in \mathscr{P}_{BPP}^{grd}$, the following hold:

$p + (q + r)$	$\sim_h$	$(p + q) + r$	(associativity)
$p + q$	$\sim_h$	$q + p$	(commutativity)
$p + \mathbf{0}$	$\sim_h$	p	(identity)
$p + p$	$\sim_h$	p	(idempotency)

PROOF. For each law, it is enough to exhibit a suitable h-team bisimulation relation on places, where each place is actually a process term, according to the net semantics. For instance, for idempotency, for each p ($p \neq \mathbf{0}$) in syntactic category s, take the relation $R_p = \{(p + p, p)\} \cup \mathscr{I}_p$ where $\mathscr{I}_p = \{(q, q) \mid q \in reach(p)\}$ is the identity relation. It is an easy exercise to check that R_p is an h-team bisimulation on the places of $[\![p + p]\!]_{\emptyset}$ and $[\![p]\!]_{\emptyset}$. In fact, if $p \xrightarrow{\mu} m$, then (according to the semantics for $p + p$) also $p + p \xrightarrow{\mu} m$ and $(m, m) \in \mathscr{I}_p^{\oplus}$, and so $(m, m) \in R_p^{\oplus}$. Symmetrically, if $p + p \xrightarrow{\mu} m$, then (according to the semantics for $p + p$) this is possible only if $p \xrightarrow{\mu} m$ is derivable and the condition $(m, m) \in R_p^{\oplus}$ is trivially satisfied. (Instead, if $p = \mathbf{0}$, then $R = \{(\mathbf{0} + \mathbf{0}, \theta)\}$ is an h-team bisimulation proving $\mathbf{0} + \mathbf{0} \sim_h \mathbf{0}$.) □

Proposition 29. (Laws of the constant) For each $p \in \mathscr{P}_{BPP}^{grd}$, and each $C \in \mathscr{C}$, the following hold:

if $C \doteq p$, then	C	$\sim_h$	p	(unfolding)
if $C \doteq p\{C/x\}$ and $q \sim_h p\{q/x\}$ then	C	$\sim_h$	q	(folding)

where, in the second law, p is actually open on x (while q is closed).

PROOF. The required h-team bisimulation on places proving the unfolding property (in case $p \neq \mathbf{0}$) is $R_{C,p} = \{(C, p)\} \cup \mathscr{I}_C$, where $\mathscr{I}_C = \{(q, q) \mid q \in reach(C)\}$ is the identity relation. (If $p = \mathbf{0}$, then the required h-team bisimulation is $\{(C, \theta)\}$.) In fact, if $C \xrightarrow{\mu} m$, then (according to the net semantics for $C \doteq p$) this means that also $p \xrightarrow{\mu} m$, with $(m, m) \in \mathscr{I}_C^{\oplus}$, and so $(m, m) \in R_{C,p}^{\oplus}$ as required. Symmetrically if p moves first.

For the folding property, note that this is implied by the following: if $q_1 \sim_h p\{q_1/x\}$ and $q_2 \sim_h p\{q_2/x\}$ then $q_1 \sim_h q_2$. In fact, if we choose $q_1 = C$, then $C = q_1 \sim_h p\{q_1/x\}$ $= p\{C/x\}$ (which holds by hypothesis, due to the unfolding property) and, moreover, $C = q_1 \sim_h q_2$, which is the thesis. This statement can be easily proven by showing that the relation $R = \{(r\{q_1/x\}, r\{q_2/x\}) \mid r \in reach(dec(p))\}$ is an h-team bisimulation up to $\sim_h$ [30]. Clearly, when $r = x$, we have that $(q_1, q_2) \in R$. So, it remains to prove the h-team bisimulation (up to) conditions. If $r\{q_1/x\} \xrightarrow{\mu} t$, this can be due to one of the following:

- $r \xrightarrow{\mu} m$ and so $t = m\{q_1/x\}$, where the substitution is applied element-wise to each place in m. In this case, also $r\{q_2/x\} \xrightarrow{\mu} m\{q_2/x\}$ is derivable such that $(m\{q_1/x\}, m\{q_2/x\}) \in R^{\oplus}$.

- $r = x$ and $q_1 \xrightarrow{\mu} m_1$, and so $t = m_1$. As $q_1 \sim_h p\{q_1/x\}$ and p is guarded, there exists m such that $p \xrightarrow{\mu} m$ and $p\{q_1/x\} \xrightarrow{\mu} m\{q_1/x\}$ with $m_1 \sim_h^{\oplus} m\{q_1/x\}$. Therefore, $p\{q_2/x\} \xrightarrow{\mu} m\{q_2/x\}$ is derivable, too. Since $q_2 \sim p\{q_2/x\}$, it follows that there exists a marking m_2 such that $q_2 \xrightarrow{\mu} m_2$ with $m_2 \sim_h^{\oplus} m\{q_2/x\}$. Summing up, if $x\{q_1/x\} = q_1 \xrightarrow{\mu} m_1$, then $x\{q_2/x\} = q_2 \xrightarrow{\mu} m_2$ such that $m_1 \sim_h^{\oplus} m\{q_1/x\}$, $(m\{q_1/x\}, m\{q_2/x\}) \in R^{\oplus}$ and, moreover, $m\{q_2/x\} \sim_h^{\oplus} m_2$, as required by the h-team bisimulation up to condition.

Simmetrically, if $r\{q_2/x\}$ moves first. Hence, R is an h-team bisimulation up to $\sim_h$. □

Proposition 30. (Laws of the parallel operator) For each $p, q, r \in \mathscr{P}_{BPP}$, the following hold:

$$\begin{array}{rcll} p\,|\,(q\,|\,r) & \sim_h^{\oplus} & (p\,|\,q)\,|\,r & \text{(associativity)} \\ p\,|\,q & \sim_h^{\oplus} & q\,|\,p & \text{(commutativity)} \\ p\,|\,\mathbf{0} & \sim_h^{\oplus} & p & \text{(identity)} \end{array}$$

PROOF. To prove that each law is sound, it is enough to observe that the net for the process in the left-hand-side is exactly the same as the net for the process in the right-hand-side. For instance, $[\![p\,|\,q]\!]_{\emptyset} = [\![q\,|\,p]\!]_{\emptyset}$. In fact, $dec(p\,|\,q) = dec(p) \oplus dec(q) = dec(q) \oplus dec(p) = dec(q\,|\,p)$ and the resulting net is obtained by simply joining the net for p with the net for q. Therefore, the identity relation on places, which is an h-team bisimulation, is enough to prove that $dec(p\,|\,q) \sim_h^{\oplus} dec(q\,|\,p)$. □

6.4. Axiomatization

In this section we provide a sound and complete, finite axiomatization of h-team bisimulation equivalence over BPP. For simplicity's sake, the syntactic definition of open BPP (cf. Definition 26) is assumed here flattened, with only one syntactic category, but we require that each ground instantiation of an axiom must respect the syntactic definition of (closed) BPP given in Section 6.1. This means that we can write the axiom $x + (y + z) = (x + y) + z$ (these terms cannot be written in open BPP according to Definition 26), but it is invalid to instantiate it to $C + (a.\mathbf{0} + b.\mathbf{0}\,|\,\mathbf{0}) = (C + a.\mathbf{0}) + (b.\mathbf{0}\,|\,\mathbf{0})$ because these are not legal BPP processes (the constant C and the parallel process $b.\mathbf{0}\,|\,\mathbf{0}$ cannot be used as summands).

The set of axioms are outlined in Table 4. We call E the set of axioms {**A1, A2, A3, A4, R1, R2, P1, P2, P3**}. By the notation $E \vdash p = q$ we mean that there exists an equational deduction proof of the equality $p = q$, by using the axioms in E. Besides the usual equational deduction rules of reflexivity, symmetry, transitivity, substitutivity and instantiation (see, e.g., [27]), in order to deal with constants we need also the following recursion congruence rule:

$$\frac{p = q \;\wedge\; A \doteq p\{A/x\} \;\wedge\; B \doteq q\{B/x\}}{A = B}$$

A1	Associativity	$x+(y+z)$	$=$	$(x+y)+z$	
A2	Commutativity	$x+y$	$=$	$y+x$	
A3	Identity	$x+\mathbf{0}$	$=$	x	
A4	Idempotence	$x+x$	$=$	x	
R1	Unfolding	if $C \doteq p$,		then	$C = p$
R2	Folding	if $C \doteq p\{C/x\} \wedge q = p\{q/x\}$,		then	$C = q$
P1	Associativity	$x\,\vert\,(y\,\vert\,z)$	$=$	$(x\,\vert\,y)\,\vert\,z$	
P2	Commutativity	$x\,\vert\,y$	$=$	$y\,\vert\,x$	
P3	Identity	$x\,\vert\,\mathbf{0}$	$=$	x	

Table 4: Axioms for h-team bisimulation equivalence

The axioms **A1-A4** are the usual axioms for choice [42]. The conditional axioms **R1-R2** are about process constants. Note that these conditional axioms are actually a finite collection of axioms, one for each constant definition: since the set $\mathscr{C}$ of process constants is finite, the instances of **R1-R2** are finitely many. Finally, we have axioms **P1-P3** for parallel composition.

Theorem 8. **(Soundness)** For every $p,q \in \mathscr{P}_{BPP}$, if $E \vdash p = q$, then $p \sim_h^{\oplus} q$.

PROOF. By induction on the proof of $E \vdash p = q$. The thesis follows by the fact that all the axioms in E are sound by Propositions 28, 29 and 30 and $\sim_h^{\oplus}$ is a congruence. $\square$

Proposition 31. **(Unique solution)** Let $\widetilde{X} = (x_1, x_2, \dots, x_n)$ be a tuple of variables and let $\widetilde{p} = (p_1, p_2, \dots, p_n)$ be a tuple of open guarded BPP terms (in syntactic category s), using the variables in $\widetilde{X}$. Then, there exists a tuple $\widetilde{q} = (q_1, q_2, \dots, q_n)$ of closed sequential BPP terms such that $E \vdash q_i = p_i\{\widetilde{q}/\widetilde{X}\}$ for $i = 1, \dots, n$. Moreover, if the same property holds for $\widetilde{q'} = (q'_1, q'_2, \dots, q'_n)$, then
$E \vdash q'_i = q_i$ for $i = 1, \dots, n$.

PROOF. By induction on n. We assume that there exists a tuple of constants $\widetilde{C} = (C_1, C_2, \dots, C_n)$ that do not occur in $\widetilde{p} = (p_1, p_2, \dots, p_n)$.

For $n = 1$, we choose $q_1 = C_1$, and we close this constant with the definition $C_1 \doteq p_1\{C_1/x_1\}$, and so the result follows immediately using axiom **R1**. This solution is unique: if $E \vdash r_1 = p_1\{r_1/x_1\}$, since $C_1 \doteq p_1\{C_1/x_1\}$, by axiom **R2** we get $E \vdash C_1 = r_1$.

Now assume a tuple $\widetilde{p} = (p_1, p_2, \dots, p_n)$ and the term p_{n+1}, so that they are all open on $\widetilde{X} = (x_1, x_2, \dots, x_n)$ and the additional x_{n+1}. Assume, w.l.o.g., that x_{n+1} occurs in p_{n+1}. First, define $C_{n+1} \doteq p_{n+1}\{C_{n+1}/x_{n+1}\}$, so that the new constant C_{n+1} is now open on $\widetilde{X}$. Therefore, also for $i = 1, \dots, n$, each $p_i\{C_{n+1}/x_{n+1}\}$ is now open on $\widetilde{X}$. Thus, we are now able to use induction on $\widetilde{X}$ and $(p_1\{C_{n+1}/x_{n+1}\}, \dots, p_n\{C_{n+1}/x_{n+1}\})$, to conclude that there exists a tuple $\widetilde{q} = (q_1, q_2, \dots, q_n)$ of closed sequential BPP terms such that

$E \vdash q_i = (p_i\{C_{n+1}/x_{n+1}\})\{\widetilde{q}/\widetilde{X}\} = p_i\{\widetilde{q}/\widetilde{X}, C_{n+1}\{\widetilde{q}/\widetilde{X}\}/x_{n+1}\}$ for $i = 1, \dots, n$.
Note that above by $C_{n+1}\{\widetilde{q}/\widetilde{X}\}$ we have implicitly closed the definition of C_{n+1} as

$C_{n+1} \doteq p_{n+1}\{C_{n+1}/x_{n+1}\}\{\widetilde{q}/\widetilde{X}\} = p_{n+1}\{\widetilde{q}/\widetilde{X}\}\{C_{n+1}/x_{n+1}\}$,

so that C_{n+1} can be chosen as q_{n+1}. By axiom **R1**, $E \vdash C_{n+1} = p_{n+1}\{\widetilde{q}/\widetilde{X}\}\{C_{n+1}/x_{n+1}\}$.

Unicity of the tuple $(\widetilde{q}, q_{n+1})$ can be proved by using axiom **R2**. Assume to have another solution tuple $(\widetilde{q'}, q'_{n+1})$. This means that

$E \vdash q'_i = p_i\{\widetilde{q'}/\widetilde{X}, q'_{n+1}/x_{n+1}\}$ for $i = 1, \dots, n+1$.

By induction, we can assume that $E \vdash q_i = q'_i$, for $i = 1, \dots, n$.

Since $E \vdash C_{n+1} = p_{n+1}\{\widetilde{q}/\widetilde{X}\}\{C_{n+1}/x_{n+1}\}$ by axiom **R1**, by substitutivity we get $E \vdash C_{n+1} = p_{n+1}\{\widetilde{q'}/\widetilde{X}\}\{C_{n+1}/x_{n+1}\}$. Let F be a constant defined as follows: $F \doteq p_{n+1}\{\widetilde{q'}/\widetilde{X}\}\{F/x_{n+1}\}$. Then, by axiom **R2**, $E \vdash C_{n+1} = F$. Hence, since

$E \vdash q'_{n+1} = p_{n+1}\{\widetilde{q'}/\widetilde{X}\}\{q'_{n+1}/x_{n+1}\}$

by axiom **R2**, we get $E \vdash F = q'_{n+1}$; so the thesis $E \vdash C_{n+1} = q'_{n+1}$ by transitivity. □

Lemma 2. For each $p \in \mathscr{P}_{BPP}$, if $p \sim_h^{\oplus} \theta$, then $E \vdash p = \mathbf{0}$.

PROOF. By induction on the structure of p. The base case is $p = \mathbf{0}$ and the thesis follows trivially. If $p = p_1 + p_2$, then $p_1 \sim_h \theta \sim_h p_2$, so that by induction $E \vdash p_i = \mathbf{0}$ for $i = 1, 2$. Hence, $E \vdash p = \mathbf{0}$ by substitutivity and axiom **A3**. If $p = C$ with $C \doteq r$, then also $r \sim_h \theta$, so that, by induction, $E \vdash r = \mathbf{0}$. Since by axiom **R1** we have $E \vdash C = r$, the thesis follows by transitivity. Finally, if $p = p_1 \,|\, p_2$, then $p_1 \sim_h^{\oplus} \theta$ and $p_2 \sim_h^{\oplus} \theta$, so that, by induction, we can derive that $E \vdash p_1 = \mathbf{0}$ and $E \vdash p_2 = \mathbf{0}$; the thesis $E \vdash p = \mathbf{0}$ follows by substitutivity and axiom **P3**. □

Proposition 32. **(Equational characterization)** If $p \in \mathscr{P}_{BPP}^{seq}$ is such that $p \nsim_h \theta$, then there exists a set $\{p_1, p_2, \dots, p_k\} \subseteq \mathscr{P}_{BPP}^{seq}$ such that $k \geq 1$, $E \vdash p = p_1$ and, for $i = 1, \dots, k$, $E \vdash p_i = p'_i$, where p'_i is of the form $\sum_{j=1}^{n(i)} a_{ij}.q_{ij}$ (with $n(i) \geq 1$) such that $dom(d(dec(q_{ij}))) \subseteq \{p_1, p_2, \dots, p_k\}$.

PROOF. The proof is by induction on the structure of p, with the proviso to use a set I of already scanned constants, in order to avoid looping on recursively defined constants, where I is initially empty. For inducing on the structure of the pair (p, I), we need to introduce an auxiliary definition: with $(E, I) \vdash p = q$ we mean that this equality is derivable by the axioms in E when each $C \in I$ is defined as $C \doteq \mathbf{0}$.

We prove that for (p, I) there exists a set $\{p_1, p_2, \dots, p_k\} \subseteq \mathscr{P}_{BPP}^{seq}$ such that $k \geq 1$, $(E, I) \vdash p = p_1$ and, for $i = 1, \dots, k$, $(E, I) \vdash p_i = p'_i$, where p'_i can be $\mathbf{0}$ (when $I \neq \emptyset$) or a sumform $\sum_{j=1}^{n(i)} a_{ij}.q_{ij}$ (with $n(i) \geq 1$) such that $dom(d(dec(q_{ij}))) \subseteq \{p_1, p_2, \dots, p_k\}$. The thesis then follows by considering $(p, \emptyset)$. In all the cases, except for the case of process constants, the parameter I is omitted for the sake of simplicity.

If $p = \mu.q$, then let $d(dec(q)) = k_1 \cdot r_1 \oplus k_2 \cdot r_2 \oplus \dots \oplus k_h \cdot r_h$. If $h = 0$ (i.e., there is not even one non-deadlock place in $dec(q)$), then $E \vdash q = \mathbf{0}$ by Lemma 2, and the thesis follows trivially by choosing $p_1 = \mu.\mathbf{0} = p'_1$. Otherwise, by induction, for each $j = 1, \dots, h$ there exist $\{r_1^j, \dots r_{l_j}^j\} \subseteq \mathscr{P}_{BPP}^{seq}$ such that $E \vdash r_j = r_1^j$, $E \vdash r_i^j = s_i^j$ for $i = 1, \dots, l_j$, where s_i^j is either $\mathbf{0}$ or a sumform $\sum_{h=1}^{n(j)} a_{ih}.t_{ih}^j$ such that $dom(d(dec(t_{ih}^j))) \subseteq$

$\{r^j_1, \dots r^j_{l_j}\}$. We choose $p_1 = \mu.t$, where t is a term such that $dec(t) = k_1 \cdot r^1_1 \oplus k_2 \cdot r^2_1 \oplus \dots \oplus k_h \cdot r^h_1$; indeed, $E \vdash p = p_1$ via axioms **P1-P3** (for reordering the sequential subterms and absorbing **0**'s), and axioms **R1** and **A1-A3** (for transforming deadlock places into **0**'s) and by substitutivity. Moreover, the set of sequential processes is $\{p_1\} \cup \{r^1_1, \dots r^1_{l_1}\} \cup \dots \cup \{r^h_1, \dots r^h_{l_h}\}$. Since for each r^j_i there is already a suitable s^j_i, it remains to define p'_1, which is $p'_1 = \mu.t$.

If $p = r_1 + r_2$, then the case $r_1 \sim_h \mathbf{0} \sim_h r_2$ is impossible, as in such a case $p \sim_h \theta$. In case $r_1 \sim_h \mathbf{0} \nsim_h r_2$, then, by induction, there exist $\{r^2_1, \dots r^2_{k_2}\}$ such that $E \vdash r_2 = r^2_1$, and, for $i = 1, \dots, k_2$, $E \vdash r^2_i = s^2_i$, where s^2_i is either $\mathbf{0}$ or a sumform $\sum_{h=1}^{n_2(i)} a_{ih}.t^2_{ih}$ such that $dom(d(dec(t^2_{ih}))) \subseteq \{r^2_1, \dots r^2_{k_2}\}$. We can take $p_i = r^2_i$, $p'_i = s^2_i$ for $i = 1, \dots, k_2$, with $E \vdash p = p_1$ by substitutivity and axioms **A2-A3** (in fact, $E \vdash r_1 = \mathbf{0}$, by Lemma 2), and $E \vdash p_i = p'_i$ by inductive assumption. Symmetrically in case $r_1 \nsim_h \mathbf{0} \sim_h r_2$. Otherwise (i.e., when $r_1 \nsim_h \mathbf{0} \nsim_h r_2$), by induction there exist $\{r^1_1, \dots r^1_{k_1}\}$ and $\{r^2_1, \dots r^2_{k_2}\}$, such that $E \vdash r_1 = r^1_1$, $E \vdash r_2 = r^2_1$, and (for $j = 1,2$) $E \vdash r^j_i = s^j_i$, where s^j_i is either $\mathbf{0}$ or a sumform $\sum_{h=1}^{n_j(i)} a_{ih}.t^j_{ih}$ such that $dom(d(dec(t^j_{ih}))) \subseteq \{r^j_1, \dots r^j_{k_j}\}$. We can take $p_1 = r^1_1 + r^2_1$ so that the set is $\{p_1\} \cup \{r^1_1, \dots r^1_{k_1}\} \cup \{r^2_1, \dots r^2_{k_2}\}$. Since for each r^j_i there is already a suitable s^j_i, it remains to define $p'_1 = s^1_1 + s^2_1 = \sum_{j=1}^{n_1(1)} a_{1j}.t^1_{1j} + \sum_{j=1}^{n_2(1)} a'_{1j}.t^2_{1j}$.

In case $p = C$, we have to consider the second parameter I: if (C, I) is such that $C \in I$, then $p_1 = C$ and $p'_1 = \mathbf{0}$. In fact, $(E, I) \vdash C = \mathbf{0}$ follows by axiom **R1**. If $C \notin I$ and $C \doteq r$, then we have to distinguish two subcases:

(i) If $r \sim_h \theta$, then this case is not possible because $C \sim_h \theta$ (excluded by hypothesis).

(ii) If $r \nsim_h \theta$, then, by induction on $(r, I \cup \{C\})$, we know that for r there exist $k \geq 1$ and $\{r_1, \dots, r_k\}$ such that $(E, I \cup \{C\}) \vdash r = r_1$ and for $i = 1, \dots, k$, $(E, I \cup \{C\}) \vdash r_i = r'_i$ where r'_i is either $\mathbf{0}$ or a sumform $\sum_{j=1}^{n(i)} a_{ij}.t_{ij}$ such that $dom(d(dec(t_{ij}))) \subseteq \{r_1, r_2, \dots, r_k\}$. Note that, by construction it follows that not only $(E, I \cup \{C\}) \vdash r = r_1$ but also $(E, I) \vdash r = r_1$. Moreover, for each i such that $r_i \neq C$, we have that not only $(E, I \cup \{C\}) \vdash r_i = r'_i$, but also $(E, I) \vdash r_i = r'_i$. Therefore, since, by axiom **R1**, $(E, I) \vdash C = r$, we have $(E, I) \vdash C = r'_1$ by transitivity, where $r'_1 = \sum_{j=1}^{n(1)} a_{1j}.t_{1j}$. Hence, we can choose $p_1 = C$ and, for $i = 1, \dots, k$, $p_{i+1} = r_i$ and, moreover, $p'_1 = r'_1$ and $p'_{i+1} = r'_i$ if $r_i \neq C$, while $p'_{i+1} = r'_1$ otherwise. Note that $(E, I) \vdash p = p_1$ (as both are C) and also that $(E, I) \vdash p_i = p'_i$ for $i = 1, \dots, k+1$. Finally, note that for each $r_i = C$, we have turned r'_i from $\mathbf{0}$ to r'_1, so that p'_{i+1} is $\mathbf{0}$ only when $r_i \in I$; therefore, we can conclude that, when $I = \emptyset$, all the p'_i's are non-empty sumforms. □

Example 16. To illustrate how induction works in the proof of the proposition above, take the constant $C \doteq a.((C \mid (\mathbf{0} + \mathbf{0})) \mid (b.\mathbf{0} + \mathbf{0}))$. We have to start with $(C, \emptyset)$, whose solution requires to consider $(a.((C \mid (\mathbf{0}+\mathbf{0})) \mid (b.\mathbf{0}+\mathbf{0})), \{C\})$, in turn requiring to consider $(C, \{C\})$ and $(b.\mathbf{0} + \mathbf{0}, \{C\})$. As $C \in \{C\}$, we have that for $(C, \{C\})$ the required terms are $p_1 = C$ and $p'_1 = \mathbf{0}$, while for $(b.\mathbf{0} + \mathbf{0}, \{C\})$ we get $p_1 = b.\mathbf{0} = p'_1$. Now we can compute the terms associated with $(a.((C \mid (\mathbf{0} + \mathbf{0})) \mid (b.\mathbf{0} + \mathbf{0})), \{C\})$, which are: $p_1 = a.(C \mid b.\mathbf{0}) = p'_1$, $p_2 = C$, $p'_2 = \mathbf{0}$, $p_3 = b.\mathbf{0} = p'_3$. So, now we can compute the terms for $(C, \emptyset)$ which are: $p_1 = C$, $p'_1 = a.(C \mid b.\mathbf{0})$, $p_2 = a.(C \mid b.\mathbf{0}) = p'_2$ $p_3 = C$, $p'_3 = p'_1$, $p_4 = b.\mathbf{0} = p'_4$. Summing up, after removing the duplicates, the required terms for C are $p_1 = C, p_2 = a.(C \mid b.\mathbf{0}), p_3 = b.\mathbf{0}$, with associated terms $p'_1 = a.(C \mid b.\mathbf{0})$, $p'_2 = a.(C \mid b.\mathbf{0})$ and $p'_3 = b.\mathbf{0}$.

As a further tiny example, consider the term $a.C + b.D$ with $C \doteq \mathbf{0}$ and $D \doteq c.d.D$. We have to start with $(a.C+b.D, \emptyset)$, whose solution requires to consider $(a.C, \emptyset)$ and $(b.D, \emptyset)$. Since $d(dec(C)) = \theta$, we have that $p_1 = a.\mathbf{0} = p'_1$. Since $d(dec(D)) = \{D\}$, we have to induce on $(D, \emptyset)$, and so on $(c.d.D, \{D\})$. This latter originates the terms $p_1 = c.d.D = p'_1$, $p_2 = d.D = p'_2$, $p_3 = D$ and $p'_3 = \mathbf{0}$, so that, for $(D, \emptyset)$, we get $p_1 = D$, $p'_1 = c.d.D$, $p_2 = c.d.D = p'_2$, $p_3 = d.D = p'_3$, $p_4 = D$ and $p'_4 = c.d.D$ (of course, p_4 is redundant and so omitted in the following). So, for $(b.D, \emptyset)$ we get the terms $p_1 = b.D = p'_1$, $p_2 = D$, $p'_2 = c.d.D$, $p_3 = c.d.D = p'_3$, $p_4 = d.D = p'_4$. Finally, for $(a.C+b.D, \emptyset)$, we get $p_1 = a.C + b.D = p'_1$, $p_2 = a.\mathbf{0} = p'_2$, $p_3 = D$, $p'_3 = c.d.D$, $p_4 = c.d.D = p'_4$, $p_5 = d.D = p'_5$. □

Proposition 33. (Completeness for sequential terms)
For each $p, p' \in \mathscr{P}_{BPP}^{seq}$, if $p \sim_h p'$, then $E \vdash p = p'$.

PROOF. If $p \sim_h \theta \sim_h p'$, then the thesis follows by Lemma 2.

Otherwise, by Proposition 32, we have that there exists a set $\{p_1, p_2, \dots, p_k\}$ of sequential processes such that $E \vdash p = p_1$, and there exist $r_1, r_2, \dots, r_k$ such that, for $i = 1, \dots, k$, $E \vdash p_i = r_i$ and r_i is a sumform $\sum_{j=1}^{n(i)} a_{ij}.t_{ij}$ (with $n(i) \geq 1$) such that $dom(d(dec(t_{ij}))) \subseteq \{p_1, p_2, \dots, p_k\}$. This means that each p_i is a non-deadlock place.

Similarly, there exists a set $\{p'_1, p'_2, \dots, p'_h\}$ of sequential processes such that $E \vdash p' = p'_1$, and there exist $r'_1, r'_2, \dots, r'_h$ such that, for $i = 1, \dots, h$, $E \vdash p'_i = r'_i$ and r'_i is a sumform $\sum_{j=1}^{n'(i)} a'_{ij}.t'_{ij}$ (with $n'(i) \geq 1$) such that $dom(d(dec(t'_{ij}))) \subseteq \{p'_1, p'_2, \dots, p'_h\}$. So, each p'_i is a non-deadlock place.

By Theorem 8, we have that $p \sim_h p_1 \sim_h r_1$ and $p' \sim_h p'_1 \sim_h r'_1$; as by hypothesis $p \sim_h p'$, by transitivity we have that $p_1 \sim_h p'_1$ and $r_1 \sim_h r'_1$. Let $r_1 = \sum_{j=1}^{n(1)} a_{1j}.t_{1j}$ and $r'_1 = \sum_{j=1}^{n'(1)} a'_{1j}.t'_{1j}$. We want to prove that $E \vDash r_1 = r'_1$ so that the thesis $E \vdash p = p'$ follows by transitivity.

Now, let $I = \{(i, i') \mid p_i \sim_h p'_{i'}\}$. Clearly, $(1,1) \in I$. As p_i and $p'_{i'}$ are h-team bisimilar when $(i, i') \in I$, the following hold: for each $(i, i') \in I$, there exists a total surjective relation $J_{ii'}$ between $\{1, 2, \dots n(i)\}$ and $\{1, 2, \dots n'(i')\}$ given by $J_{ii'} = \{(j, j') \mid a_{ij} = a'_{i'j'} \wedge (d(dec(t_{ij})), d(dec(t'_{i'j'}))) \in I^{\oplus}\}$, where $(d(dec(t_{ij})), d(dec(t'_{i'j'}))) \in I^{\oplus}$ if

- $d(dec(t_{ij})) = p_{d_{(i,j,1)}} \oplus p_{d_{(i,j,2)}} \oplus \dots \oplus p_{d_{(i,j,n)}}$ such that $1 \leq d_{(i,j,l)} \leq k$ for $l = 1, \dots, n$;
- $d(dec(t'_{i'j'})) = p'_{d'_{(i',j',1)}} \oplus p'_{d'_{(i',j',2)}} \oplus \dots \oplus p'_{d'_{(i',j',n)}}$, such that $1 \leq d'_{(i',j',l)} \leq h$ for $l = 1, \dots, n$; and
- $(d_{(i,j,l)}, d'_{(i',j',l)}) \in I$ for $l = 1, \dots, n$. (If $n = 0$, then $(\theta, \theta) \in I^{\oplus}$).

Now, for each $(i, i') \in I$, let us consider the variables $x_{ii'}$ and the open term

$$v_{ii'} = \sum_{(j,j') \in J_{ii'}} a_{ij}.(x_{d_{(i,j,1)} d'_{(i',j',1)}} \mid x_{d_{(i,j,2)} d'_{(i',j',2)}} \mid \dots \mid x_{d_{(i,j,n)} d'_{(i',j',n)}})$$

where, if $J_{ii'} = \emptyset$ then $v_{ii'} = \mathbf{0}$, while in case $n = 0$, the open parallel process
$x_{d_{(i,j,1)}d'_{(i',j',1)}} \,|\, x_{d_{(i,j,2)}d'_{(i',j',2)}} \,|\, \dots \,|\, x_{d_{(i,j,n)}d'_{(i',j',n)}}$
is actually $\mathbf{0}$. By Proposition 31, for each $(i,i') \in I$, there exists $s_{ii'}$ such that $E \vdash s_{ii'} = v_{ii'}\{\widetilde{s}/\widetilde{X}\}$, where $\widetilde{s}$ denotes the tuple of terms of the form $s_{ii'}$ for each $(i,i') \in I$, and $\widetilde{X}$ denotes the tuple of variables $x_{ii'}$ for each $(i,i') \in I$.

If we close each $v_{ii'}$ by replacing each $x_{d_{(i,j,l)}d'_{(i',j',l)}}$ with $p_{d_{(i,j,l)}}$, we get

$$\sum_{(j,j') \in J_{ii'}} a_{ij}.(p_{d_{(i,j,1)}} \,|\, p_{d_{(i,j,2)}} \,|\, \dots \,|\, p_{d_{(i,j,n)}})$$

which is equal, up to axioms **P1-P3** (for reordering the sequential subterms and absorbing **0**'s) and axioms **R1** and **A1-A3** (for transforming deadlock places into **0**'s), to $\sum_{(j,j') \in J_{ii'}} a_{ij}.t_{ij}$, in turn equal, via axioms **A1-A4**, to r_i: in fact, $J_{ii'}$ is surjective so that the two summations differ only for possible repeated summands. Since $E \vdash p_i = r_i$ for $i = 1,\dots,k$, we get
$E \vdash r_i = \sum_{(j,j') \in J_{ii'}} a_{ij}.(r_{d_{(i,j,1)}} \,|\, r_{d_{(i,j,2)}} \,|\, \dots \,|\, r_{d_{(i,j,n)}})$.
Therefore, we note that r_i is such that $E \vdash r_i = v_{ii'}\{\widetilde{r}/\widetilde{X}\}$ and so, by Proposition 31, we have that $E \vdash s_{ii'} = r_i$. Since $(1,1) \in I$, we have that $E \vdash s_{11} = r_1$.

Similarly, if we close each $v_{ii'}$ by replacing each $x_{d_{(i,j,l)}d'_{(i',j',l)}}$ with $p'_{d'_{(i',j',l)}}$, we get

$$\sum_{(j,j') \in J_{ii'}} a_{ij}.(p'_{d'_{(i',j',1)}} \,|\, p'_{d'_{(i',j',2)}} \,|\, \dots \,|\, p'_{d'_{(i',j',n)}})$$

which is equal, up to axioms **P1-P3** (for reordering the sequential subterms and absorbing **0**'s) and axioms **R1** and **A1-A3** (for transforming deadlock places into **0**'s), to $\sum_{(j,j') \in J_{ii'}} a_{ij}.t'_{i'j'}$, in turn equal, via axioms **A1-A4**, to $r'_{i'}$: in fact, $J_{ii'}$ is surjective so that the two summations differ only for possible repeated summands. Since $E \vdash p'_i = r'_i$ for $i = 1,\dots,h$, we get
$E \vdash r'_{i'} = \sum_{(j,j') \in J_{ii'}} a_{ij}.(r'_{d'_{(i',j',1)}} \,|\, r'_{d'_{(i',j',2)}} \,|\, \dots \,|\, r'_{d'_{(i',j',n)}})$.
Thus, we note that $r'_{i'}$ is such that $E \vdash r'_{i'} = v_{ii'}\{\widetilde{r'}/\widetilde{X}\}$ and so, by Proposition 31, we have that $E \vdash s_{ii'} = r'_{i'}$. Since $(1,1) \in I$, we have that $E \vdash s_{11} = r'_1$; by transitivity, it follows that $E \vdash r_1 = r'_1$, and so that $E \vdash p = p'$. □

Theorem 9. (Completeness) For every $p,q \in \mathscr{P}_{BPP}$, if $p \sim_h^{\oplus} q$, then $E \vdash p = q$.

PROOF. The proof is by induction on the sum of the sizes of $dec(p)$ and $dec(q)$. If $|dec(p)| + |dec(q)| = 0$, then $dec(p) = \theta = dec(q)$. By observing the definition of the decomposition function in Table 2, this is possible only if p and q are either **0** or a parallel composition of **0**'s, e.g., $\mathbf{0}\,|\,\mathbf{0}$; hence, $E \vdash p = \mathbf{0}$ and $E \vdash q = \mathbf{0}$, possibly using axioms **P1-P3**; hence, by transitivity we get $E \vdash p = q$.

If $|dec(p)| + |dec(q)| = k+1$, then we can assume, w.l.o.g., that $|dec(p)| \geq 1$, so that there exist p_1 and p_2 such that $dec(p) = p_1 \oplus dec(p_2) = dec(p_1 \,|\, p_2)$, so that $E \vdash p = p_1 \,|\, p_2$ by axioms **P1-P3**.

If $p_1 \sim_h \theta$, then $dec(p) \sim_h^{\oplus} dec(p_2)$ and, by Lemma 2, we get $E \vdash p_1 = \mathbf{0}$. Hence, $E \vdash p = p_2$ by substitutivity and axiom **P3**. Since $dec(p) \sim_h^{\oplus} dec(q)$, by transitivity we get $dec(p_2) \sim_h^{\oplus} dec(q)$, so that, by induction, we get $E \vdash p_2 = q$. Thus, the thesis $E \vdash p = q$ follows by transitivity.

Otherwise, if $p_1 \nsim_h \theta$, then q_1, q_2 exist such that $p_1 \sim_h q_1$, $dec(p_2) \sim_h^{\oplus} dec(q_2)$ and $dec(q) = q_1 \oplus dec(q_2)$. By the definition of the decomposition function and by axioms **P1-P3**, this means that $E \vdash p = p_1 \mid p_2$ and $E \vdash q = q_1 \mid q_2$. By Proposition 33 we have that $E \vdash p_1 = q_1$. By induction, we have that $E \vdash p_2 = q_2$. By substitutivity we get $E \vdash p_1 \mid p_2 = q_1 \mid q_2$ and so the thesis follows by transitivity. □

7. Conclusion, Related Literature and Future Research

Team bisimulation equivalence is a truly concurrent equivalence which is the most natural, intuitive and simple extension of LTS bisimulation equivalence to BPP nets. It also has a rather low complexity: indeed, by adapting the Kanellakis-Smolka algorithm [37, 38], $\sim$ can be computed in $O(m \cdot p^2 \cdot n)$ time, where m is the number of net transitions, p is the size of the largest post-set (i.e., p is the least natural such that $|t^{\bullet}| \leq p$ for all t) and n is the number of places (or in $O(m \cdot n^2)$, *cf.* Remark 5). After having computed $\sim$, checking whether two markings of size k are team bisimilar can be done in $O(k^2)$ (or in $O(n)$, *cf.* Remark 5). On the contrary, interleaving bisimilarity over BPP nets is PSPACE-complete [35]. As, in order to perform team bisimulation equivalence checking, there is no need to compute the LTSs of the global behavior of the systems under scrutiny, our proposal seems a natural solution to solving the state-space explosion problem for BPP nets.

Future work may be devoted to see whether the Paige-Tarjan algorithm [50] (which is more performant than the Kanellakis-Smolka one, as it computes bisimilarity on an LTS with n states and m transitions in $O(m \cdot log\, n)$) can be adapted to compute (h-)team bisimilarity on BPP nets in a more efficient manner.

Team bisimulation equivalence coincides with causal-net bisimilarity and state-sensitive fully-concurrent bisimilarity (Corollaries 2 and 3), hence it corresponds to the intuitively correct bisimulation-based causal semantics for BPP nets. Moreover, it also coincides (as proved in [30]) with *strong place bisimilarity* [3, 4] and with *structure-preserving bisimilarity*, because our causal-net bisimilarity is similar to the process-oriented characterization of that equivalence in [24]. From a technical point of view, team bisimulation seems a sort of *egg of Columbus*: a simple (actually, a bit surprising in its simplicity) solution for a presumedly hard problem.

This paper is not only an addition to [30], where team bisimilarity was originally introduced, but also an extension to a team-style characterization of fully-concurrent bisimilarity, namely h-team bisimilarity. For this variant team bisimulation equivalence, we have described a modal logic characterization and a finite, sound and complete, axiomatization; therefore, we have characterized fully-concurrent bisimilarity for BPP nets logically and axiomatically.

The complexity results we have obtained for fully-concurrent bisimilarity (cf. Remark 8), by means of its characterization in terms of h-team bisimilarity, seem roughly comparable with those in [20], where, by using an event structure [53] semantics, Fröschle et al. show that history-preserving bisimilarity is decidable for the BPP process algebra with guarded summation in $O(n^3 \cdot \log n)$ time, where n is the *size of the* *involved BPP terms*; however, this value n – defined as "the total number of occurrences of symbols (including parentheses)" [20] – is much larger than the size of the corresponding BPP net, and so our complexity result seems better.

We think that causal-net bisimilarity and state-sensitive fc-bisimilarity (hence, also team bisimilarity) are more accurate than fc-bisimilarity (hence, h-team bisimilarity) because they are *resource-aware*. In the implementation of a system, a token is an instance of a sequential process, so that a processor is needed to execute it. If two markings have different size, then a different number of processors is necessary. Hence, a semantics such as causal-net bisimilarity, which relates only markings of the same size, is more accurate as it equates distributed systems only if they require the same amount of execution resources. Paper [32] offers, in the area of information flow security, further arguments in favor of these concrete equivalences. Moreover, [24] argues that the resource-aware *structure-preserving bisimilarity* (which coincides with team bisimilarity on BPP nets) is the coarsest semantics respecting *inevitability* [44], i.e., if two systems are equivalent, and in one the occurrence of a certain action is inevitable, then so is it in the other one.

The modal logic HTML extends HML [33, 2] with an operator $\otimes$ of parallel composition of formulae, in the style of Caires' and Cardelli's *spatial logic* [9]. In order to characterize team bisimilarity, we proposed in [30] a slightly more discriminating modal logic, called TML, which is sensitive to the kind of termination. To this aim, TML exploits two atomic propositions *nn* and *vv*, such that *nn* is satisfied by all the places, while *vv* is satisfied by θ only. Therefore, the empty marking and a deadlock place are distinguished by this logic. On the contrary, the semantic definition of HTML (as well as of HML) is insensitive to the kind of termination: a deadlock place and the empty marking satisfy the same HTML formulae. This difference between the semantics of TML and that of HTML is the essence of the difference between team bisimulation equivalence and h-team bisimulation equivalence. Hence, the modal logic HTML proposed here provides an extremely simple modal logic characterization of h-team equivalence (and so also of fully-concurrent bisimilarity) over BPP nets. More complex modal logics characterizing some non-interleaving equivalences on larger classes of distributed systems have been proposed in, e.g., [8, 5]. A possible future work is to extend HTML to become a temporal logic, with least and greatest fixpoint operators, as in Kozen's modal mu-calculus [39].

Since BPP (with guarded sum) is the process algebra representing, up to net isomorphism, all the possible BPP nets [28], it is interesting to compare team (and h-team) bisimulation equivalence with other non-interleaving equivalences proposed on process algebras, such as *causal bisimilarity* [13], *distributed bisimilarity* [10], *location bisimilarity* [11] or *performance bisimilarity* [26]. On BPP (with guarded sum) all these non-interleaving equivalences do coincide [1, 18, 40] with *history-preserving bisimilarity* [51, 22, 14]. As discussed above, team bisimulation equivalence coincides with causal-net bisimilarity, and so it is slightly finer than history-preserving bisimilarity, which coincides with fully-concurrent bisimilarity (hence with h-team bisimilarity). For instance, let C be a process constant with empty body, $C \doteq \mathbf{0}$; then the two BPP process terms $a.\mathbf{0}$ and $a.\mathbf{0} \,|\, C$ are history-preserving bisimilar, but they generate two markings of different size, and so they are not related by team bisimulation equivalence.

The axiomatization we have provided here for h-team bisimulation equivalence is an adaptation of the axiomatization for team bisimulation equivalence outlined in [30]. The main difference is that in [30] axioms **A3-A4** have the side condition $x \neq \mathbf{0}$ and that axiom **R1** has the side condition $p \neq \mathbf{0}$ (and so there is one extra axiom handling the case when $C \doteq \mathbf{0}$: in such a case $C = \mathbf{0} + \mathbf{0}$). These side-conditions are necessary as the net semantics for $\mathbf{0}$ is the empty marking θ, while the net semantics for $\mathbf{0} + \mathbf{0}$ (as well as of $C \doteq \mathbf{0}$) is a deadlock place, which is not team bisimilar to θ.

Our axiomatization, and the proof techniques we adopted to prove its completeness, are based on [42], where Milner provided a finite axiomatization of interleaving bisimilarity for finite-state CCS. Nonetheless, our technical treatment, based on constants defined over *guarded* processes (e.g., $C \doteq a.C$) rather than on the recursive operator (with possible *unguarded* variables; e.g. $fixX.(a.X + X)$), is a bit simpler than that. In fact, the axiomatization in [42] uses one further axiom for handling unguardedness: $fixX.(p+X) = fixX.p$.

In the literature there is a sort of dichotomy between the use of process constants or of the fixpoint operator for expressing recursive behavior. Indeed, recursion by constants is considered more useful for applications (see, e.g., [43], page 56), but for theoretical studies the fixpoint operator is considered more convenient. To confirm this view, Milner wrote ([43] on page 165):

> "For formal studies such as proof of completeness it is preferable not to admit Constants with defining equations, but to stick to the pure $fixX\_$ expressions introduced in Section 2.9 - even though they are not very pleasant to use in applications."

Actually, we disproved his statement by showing that recursion by process constants can be treated technically in a very satisfactory manner also for axiomatizations. Moreover, the use of process constant allowed us to define, in a very simple way, a denotational net semantics for BPP (cf. Table 3); on the contrary, a denotational net semantics for the BPP variant with the fixpoint operator is not immediate.

To the best of our knowledge, in the literature there are only other two examples of *finite* axiomatizations for truly-concurrent equivalences on BPP. In [12] *distributed bisimilarity* [10] is axiomatized for BPP with guarded summation, and in [19] *hereditary history-preserving bisimilarity* [36] is axiomatized for full BPP with a sequent-based approach. These two finite axiomatizations are actually sound and complete also for *history-preserving bisimilarity* (hpb, for short), as on BPP with guarded summation these two behavioral equivalences coincide with hpb [1, 18]. Our axiomatization is an alternative finite axiomatization of hpb for BPP (with guarded summation and guarded recursion).

Interestingly, the modal characterization (i.e., HTML) and the axiomatization of h-team bisimilarity (cf. Table 4) for BPP are conservative extensions of the corresponding modal (i.e. HML) and axiomatic (i.e., axioms **A1-A4** and **R1-R2**) characterizations of interleaving bisimulation on finite-state CCS [42, 33, 43]. Indeed, we think that our contribution sheds light on the fact that BPP h-team bisimilarity seems the most natural conservative extension of finite-state CCS interleaving bisimulation to the distributed setting where systems are composed of a collection of sequential (but that can fork, so that this collection can grow unboundedly), non-cooperating processes.

The linear-time variants of history-preserving bisimilarity and causal-net bisimilarity are partial-order-trace equivalence (Definition 15) and causal-trace equivalence (Definition 14), respectively. On the BPP process algebra, we have that, e.g., $p = a.(b.\mathbf{0} + c.\mathbf{0}) \,|\, a.d.\mathbf{0}$ generates the same causal nets as $q = a.d.\mathbf{0} \,|\, (a.b.\mathbf{0} + a.c.\mathbf{0})$, i.e., $p =_{ct} q$, and also that $p' = a.(b.(\mathbf{0}+\mathbf{0}) + c.\mathbf{0}) \,|\, a.d.(\mathbf{0}+\mathbf{0})$ generates isomorphic partial orders to those of q, i.e., $p' =_{pt} q$. We conjecture that these linear-time equivalences can be axiomatized for BPP, by simply adding the distributivity axiom

$\mu.(x+y) = \mu.x + \mu.y$

to the axiomatization of h-team bisimilarity in Table 4 and the corresponding axiom table for team bisimilarity in [30].

As future research, we plan to investigate *weak* team/h-team bisimilarity and *branching* team/h-team bisimilarity for BPP nets, following the intuition of weak bisimilarity [43] and branching bisimilarity [23] on LTSs. A first step in this direction is [31].

Acknowledgements: The anonymous reviewers are thanked for their useful comments and suggestions.

References

[1] L. Aceto, Relating distributed, temporal and causal observations of simple processes, *Funda. Info.* 17(4):319-331, 1992.

[2] L. Aceto, A. Ingólfsdóttir, K. Larsen, J. Srba, *Reactive Systems: Modelling, Specification and Verification*, Cambridge University Press, 2007.

[3] C. Autant, Z. Belmesk, Ph. Schnoebelen, Strong bisimilarity on nets revisited, in Procs. PARLE'91, vol. II: Parallel Languages, LNCS 506, 295-312, Springer, 1991.

[4] C. Autant, Ph. Schnoebelen, Place bisimulations in Petri nets, in Procs. Application and Theory of Petri Nets 1992, LNCS 616, 45-61, Springer, 1992.

[5] P. Baldan, S. Crafa, A logic for true concurrency, *J. ACM* 61(4): 24:1-24:36, 2014.

[6] E. Best, R. Devillers, Sequential and concurrent behavior in Petri net theory, *Theoretical Computer Science* 55(1):87-136, 1987.

[7] E. Best, R. Devillers, A. Kiehn, L. Pomello, Concurrent bisimulations in Petri nets, *Acta Inf.* 28(3): 231-264, 1991.

[8] J.C. Bradfield, S.B. Fröschle, Independence-friendly modal logic and true concurrency, *Nord. J. Comput.* 9(1): 102-117, 2002.

[9] L. Caires, L. Cardelli, A spatial logic for concurrency (part I), *Inf. Comput.* 186(2): 194-235, 2003.

[10] I. Castellani, M. Hennessy, Distributed bisimulations, *J. of the ACM* 36(4):887-911, 1989.

[11] I. Castellani, Process Algebras with Localities, *Handbook of Process Algebra* (J.A. Bergstra, A. Ponse, S.A. Smolka (eds.)), 945-1046, Elsevier, 2001.

[12] S. Christensen, *Decidability and Decomposition in Process Algebra*, Ph.D. Thesis, University of Edinburgh, 1993.

[13] Ph. Darondeau, P. Degano, Causal trees, in Procs. ICALP'89, LNCS 372, 234-248, Springer, 1989.

[14] P. Degano, R. De Nicola, U. Montanari, Partial ordering descriptions and observations of nondeterministic concurrent systems, in (J. W. de Bakker, W. P. de Roever, G. Rozenberg, Eds.) *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, LNCS 354, 438-466, Springer, 1989.

[15] J. Desel, W. Reisig, Place/Transition Petri nets, in *Lectures on Petri Nets I: Basic Models* Advances in Petri Nets, LNCS 1491, 122-173, Springer, 1998.

[16] J. Engelfriet, Branching processes of Petri nets, *Acta Informatica* 28(6):575-591, 1991.

[17] J. Esparza, A. Kiehn, On the model checking problem for branching-time logics and Basic Parallel Processes, in Procs. CAV'95, LNCS 939, Springer, 353-366, 1995.

[18] S. Fröschle, Decidability of plain and hereditary history-preserving bisimulation for BPP, in Procs. EXPRESS'99, ENTCS 27, 1999.

[19] S. Fröschle, S. Lasota, Decomposition and complexity of hereditary history preserving bisimulation on BPP, in Procs. CONCUR'05, LNCS 3656, 263-277, Springer, 2005.

[20] S. Fröschle, P. Jančar, S. Lasota, Z. Sawa, Non-interleaving bisimulation equivalences on Basic Parallel Processes, *Information and Computation* 208(1):42-62, 2010.

[21] R.J. van Glabbeek, F.W. Vaandrager, Petri net models for algebraic theories of concurrency, In Proc. PARLE'87, LNCS 259, 224-242, Springer, 1987.

[22] R.J. van Glabbeek, U. Goltz, Equivalence notions for concurrent systems and refinement of actions, in Procs. MFCS'89, LNCS 379, 237-248, Springer, 1989.

[23] R.J. van Glabbeek, W.P. Weijland, Branching time and abstraction in bisimulation semantics, *Journal of the ACM* 43(3):555-600, 1996.

[24] R.J. van Glabbeek, Structure preserving bisimilarity - Supporting an operational Petri net semantics of CCSP, in (R. Meyer, A. Platzer, H. Wehrheim, Eds.) *Correct System Design* — Symposium in Honor of Ernst-Rüdiger Olderog on the Occasion of His 60th Birthday, LNCS 9360, 99-130, Springer, 2015.

[25] U. Goltz, W. Reisig, The non-sequential behaviour of Petri nets, *Information and Control* 57(2-3):125-147, 1983.

[26] R. Gorrieri, M. Roccetti, E. Stancampiano, A theory of processes with durational actions, *Theoretical Computer Science* 140(1):73-94, 1995.

[27] R. Gorrieri, C. Versari, *Introduction to Concurrency Theory: Transition Systems and CCS*, EATCS Texts in Theoretical Computer Science, Springer, 2015.

[28] R. Gorrieri, *Process Algebras for Petri Nets: The Alphabetization of Distributed Systems*, EATCS Monographs in Theoretical Computer Science, Springer, 2017.

[29] R. Gorrieri, Verification of finite-state machines: A distributed approach, *Journal of Logic and Algebraic Methods in Programming* 96:65-80, 2018.

[30] R. Gorrieri, Team bisimilarity, and its associated modal logic, for BPP nets, *Acta Informatica*, 2020, DOI: 10.1007/s00236-020-00377-4

[31] R. Gorrieri, Team equivalences for finite-state machines with silent moves, *Information and Computation*, 275:104603, 2020. DOI:10.1016/j.ic.2020.104603

[32] R. Gorrieri, Interleaving vs true concurrency: some instructive security examples, in Procs. Petri Nets 2020, LNCS 12152, 131-152, Springer, 2020.

[33] M. Hennessy, R. Milner, Algebraic laws for nondeterminism and concurrency, *Journal of the ACM* 32(1):137-161, 1985.

[34] J.E. Hopcroft, R.M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM Journal on Computing*, 2 (4): 225-231,1973.

[35] P. Jančar, Strong bisimilarity on Basic Parallel Processes is PSPACE-complete, in Procs. of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03), 218-227, IEEE Computer Society Press (2003)

[36] A. Joyal, M. Nielsen, G. Winskel, Bisimulation from open maps, *Information and Computation* 127:164-185, 1996.

[37] P. Kanellakis, S. Smolka, CCS expressions, finite state processes, and three problems of equivalence, in Procs. 2nd Annual ACM Symposium on Principles of Distributed Computing, 228-240, ACM Press, 1983.

[38] P. Kanellakis, S. Smolka, CCS expressions, finite-state processes and three problems of equivalence, *Information and Computation* 86:43-68, 1990.

[39] D. Kozen, Results on the propositional mu-calculus, *Theor. Computer Science* 27:333-354, 1983.

[40] S. Lasota, Decidability of performance equivalence for Basic Parallel Processes, *Theoretical Computer Science* 360(1-3):172-192, 2006.

[41] A. Liberato, *A Study on Bisimulation Equivalence and Team Equivalence* Master Thesis of the University of Bologna (supervisor R. Gorrieri), October 2019.

[42] R. Milner, A complete inference systems for a class of regular behaviors, *J. Comput. System Sci.* 28: 439-466, 1984.

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

[44] A.W. Mazurkiewicz, E. Ochmanski, W. Penczek, Concurrent systems and inevitability, *Theoretical Computer Science* 64(3):281-304, 1989.

[45] M. Nielsen, G.D. Plotkin, G. Winskel, Petri nets, event structures and domains (part I), *Theor. Comp. Scie.* 13(1):85-108, 1981.

[46] C.A. Petri , Non-sequential processes, Internal Report GMD-ISF-77.05, GMD, St. Augustin, 1977.

[47] E.R. Olderog, *Nets, Terms and Formulas*, Cambridge Tracts in Theoretical Computer Science 23, Cambridge University Press, 1991.

[48] D.M.R. Park, Concurrency and automata on infinite sequences, In Proc. 5th GI-Conference on Theoretical Computer Science, LNCS 104, 167-183, Springer, 1981.

[49] J.L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.

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

[51] A. Rabinovich, B.A. Trakhtenbrot, Behavior structures and nets, *Fundamenta Informaticae* 11(4):357-404, 1988.

[52] W. Reisig, *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*, Springer, 2013.

[53] G. Winskel, Event structures, *Advances in Petri Nets, Part II*, Proceedings of an Advanced Course, Bad Honnef, 1986, LNCS 255, 325-392, Springer, 1987.