Symbolic Semantics Revisited*

Filippo Bonchi and Ugo Montanari

Dipartimento di Informatica, Università di Pisa

Abstract. *Symbolic bisimulations* were introduced as a mean to define value-passing process calculi using smaller, possibly finite labelled transition systems, equipped with symbolic actions. Similar ideas have been used for modeling with fewer transitions the input behavior of open and asynchronous π-calculus. In this paper we generalize the symbolic technique and apply the resulting theory to these two cases, re-deriving existing results. We also apply our approach to a new setting, i.e. *open Petri nets*, with the usual result of reducing input transitions. Our theory generalizes Leifer and Milner reactive systems by adding observations.

1 Introduction

A compositional interactive system is usually defined as a labelled transition system (LTS) where states are equipped with an algebraic structure. Behavioural equivalence is often defined as bisimilarity, namely the largest bisimulation. Then a key property is that bisimilarity be a congruence, i.e. that abstract semantics respects the algebraic operations.

When this is not the case for some operations, the obvious fix is to define the abstract semantics as the largest bisimulation which is closed for those operations. An equivalent approach is to introduce additional moves of the form $p \xrightarrow{c,a} q$, for every context c built with the faulty operations, whenever $c(p) \xrightarrow{a} q$ is a transition in the original LTS. If we call *saturated* the resulting LTS, we have that ordinary bisimilarity on the saturated LTS coincides with the largest bisimulation (on the original LTS) which is closed for the faulty operations. By analogy we call the latter *saturated bisimilarity.*

This idea was originally introduced by the second author and Sassone in [18]. They define *dynamic bisimilarity* in order to make weak bisimilarity of CCS [14] a congruence w.r.t. non-deterministic choices: before any transition, the observer inserts the processes into all possible choice contexts. Analogously, since late and early bisimilarity of π-calculus [16] are not preserved under substitution (and thus under input prefixes), in [21] Sangiorgi introduces *open bisimilarity* ($\sim^o$) as the largest bisimulation on π-calculus agents which is closed under substitutions.

Another example of saturated bisimilarity is $\sim^1$ [1] for the asynchronous π-calculus [1,9]. Here the basic bisimilarity, namely $o\tau$-bisimilarity, is not a congruence under parallel outputs, and thus at any step of definition of $\sim^1$ the observer

* Research partially supported by the IST 2004-16004 SEnSOria, and the MIUR PRIN 2005015824 ART.

R. Amadio (Ed.): FOSSACS 2008, LNCS 4962, pp. 395–412, 2008.

inserts the process in parallel with all possible outputs. In the same way, $\sim^N$ has been defined in [3] amongst open Petri nets [10,2,12] that are an interactive version of P/T Petri nets.

Here we introduce a general model encompassing the three examples above.

The definition of saturated bisimilarity as ordinary bisimulation on the saturated LTS, while in principle operational, often makes infinite state the portion of LTS reachable by any nontrivial agent, and in any case is very inefficient, since it introduces a large number of additional transitions.

Inspired by Hennessy and Lin [8], who introduced a *symbolic* semantics of value passing calculi, Sangiorgi defines in [21] a symbolic transition system and a new notion of bisimilarity that coincides with $\sim^O$. Analogously in [1], Amadio et al. defined asynchronous bisimilarity that coincides with $\sim^1$.

We define symbolic transition system and symbolic bisimilarity and we show that the latter coincides with saturated bisimilarity. We also show that the results by Sangiorgi and Amadio et al. are special cases. Concerning Petri nets, no symbolic semantics exists. Here our framework produces a new symbolic semantics (equivalent to the standard $\sim^N$) by reducing input transitions.

Our construction employs some general knowledge about the modeled formalism. For instance, we know that in π-calculus (without mismatch):

$$\text{“}\forall \text{ process } p \text{ and substitution } \sigma, \text{ if } p \xrightarrow{\mu} q \text{ then } \sigma(p) \xrightarrow{\sigma(\mu)} \sigma(q) \text{ ” [16].}$$

Thus, if in the saturated LTS, $p \xrightarrow{\phi,\mu} p'$ (meaning that $\phi(p) \xrightarrow{\mu} p'$), then surely also $p \xrightarrow{\psi(\phi),\psi(\mu)} \psi(p')$. The second transition is to some extent *redundant*, i.e., we can ignore it without changing the saturated bisimilarity. For any formalism, we identify a set of rules (given in a fixed format) expressing how contexts modify transitions and we prune the saturated LTS by employing these rules.

Our results have been inspired also by the theory of reactive systems by Leifer and Milner [11]. Their aim is to take a transition system (expressed through reaction rules) without any observations and to (automatically) derive a labeled transition system in such a way that bisimilarity is a congruence. The idea is to take as labels the *minimal* contexts that enable a reaction.

Reactive system theory has been applied to several interesting formalisms, but only rarely the canonical abstract semantics have been retrieved (amongst these, CCS in [4] and Petri nets [15,22]). In our opinion, labels cannot represent both *interactions* and *observations*, because these two concepts are different, like for example, in the asynchronous calculi where the input interaction is not observable. Thus we believe that some notion of observation, either on transitions or on states (e.g. barbs [17,20]), is necessary. In this sense we can say that our theory generalizes reactive systems by adding observations. The special case of reactive systems can be retrieved from our approach by starting from an unlabeled transition system with the rule:

$$\text{“}\forall \text{ process } p \text{ and reactive context } d, \text{ if } p \longrightarrow q \text{ then } d(p) \longrightarrow d(q)\text{”.}$$

In Sec. 2, we recall open and asynchronous bisimilarities of π-calculus. In Sec. 3, we introduce our theory and, in Sec. 4, we apply it to π-calculus. In Sec. 5, we introduce open Petri nets and, applying our approach, we get a new symbolic semantics. In Sec. 6 we show how our theory generalizes reactive systems and, in Sec. 7, we outline conclusions and future works.

2 Background on π-Calculus

Let $\mathcal{N}$ be a set of *names* (ranged over by $a, b, c \ldots$) with $\tau \notin \mathcal{N}$. The set of π-processes is defined by the following grammar:

$$p ::= \mathbf{0},\ \ \alpha.p,\ \ [a=b]p,\ \ p_1 \mid p_2,\ \ p_1 + p_2,\ \ \nu a.p,\ \ !p, \qquad \alpha ::= a(b),\ \ \overline{a}b,\ \ \tau$$

Considering $a(b).p$ and $\nu b.p$, the occurrences of b in p are bound. An occurrence of a name in a process is *free*, if it is not bound. The set of *free names* of p (denoted by $\mathrm{fn}(p)$) is the set of names that have a free occurrence in the process p. The process p is α-equivalent to q (written $p \equiv_\alpha q$), if they are equivalent up to α-renaming of bound occurrences of names. The operational semantics of π-calculus is a transition system labeled on actions $Act = \{a(b), \overline{a}b, \overline{a}(b), \tau \mid a, b \in \mathcal{N}\}$ (ranged over by μ) where b is a *bound name* (written $b \in \mathrm{bn}(\mu)$) in $a(b)$ and $\overline{a}(b)$. In all the other cases a and b are free in μ ($a, b \in \mathrm{fn}(\mu)$). By $\mathrm{nm}(\mu)$ we denote the set of both free and bound names of μ. The same notation will be used later for match sequences, distinctions and substitutions.

Table 1. Late operational semantics of π-calculus

(PRE) $\alpha.p \xrightarrow{\alpha} p$	(COM) $\dfrac{p \xrightarrow{\overline{a}b} p' \quad q \xrightarrow{a(x)} q'}{p \mid q \xrightarrow{\tau} p' \mid q'\{^b/_x\}}$	(PAR) $\dfrac{p \xrightarrow{\mu} p'}{p \mid q \xrightarrow{\mu} p' \mid q}$ $\mathrm{bn}(\mu) \cap \mathrm{fn}(q) = \varnothing$
(SUM) $\dfrac{p \xrightarrow{\mu} p'}{p + q \xrightarrow{\mu} p'}$	(OPN) $\dfrac{p \xrightarrow{\overline{a}b} p'}{\nu b.p \xrightarrow{\overline{a}(b)} p'}$ $b \neq a$	(RES) $\dfrac{p \xrightarrow{\mu} p'}{\nu b.p \xrightarrow{\mu} \nu b.p'}$ $b \notin \mathrm{nm}(\mu)$
(REP) $\dfrac{p \mid p! \xrightarrow{\mu} q}{p! \xrightarrow{\mu} q}$	(MAT) $\dfrac{p \xrightarrow{\mu} p'}{[a=a]p \xrightarrow{\mu} p'}$	(CLS) $\dfrac{p \xrightarrow{\overline{a}(x)} p' \quad q \xrightarrow{a(x)} q'}{p \mid q \xrightarrow{\tau} \nu x.p' \mid q'}$

The (late) labeled transition system is inductively defined by the rules in Table 1, where we have omitted the symmetric version of the rules SUM, PAR, COM and CLS and where we consider processes up to α-equivalence.

Open Bisimilarity. In [16], the authors introduce late and early bisimilarities. These are congruences w.r.t. parallel composition, but they are not preserved by the input prefixes. Consider the processes $p = \overline{a}b \mid c(x)$ and $q = \overline{a}b.c(x) + c(y).\overline{a}b$ (here and in the following we abbreviate $\alpha.\mathbf{0}$ with α). These are (late and early) bisimilar, but whenever we put them into the context $d(a).-$, they are not anymore. Indeed if this prefix receives c, then $a = c$, and thus p can perform a τ action (synchronizing the two parallel components), while q cannot.

Sangiorgi in [21] introduces open bisimilarity ($\sim^O$) that is a congruence with respect to all the operators. In $\sim^O$, name instantiation is part of the coinductive definition of bisimilarity. At any step of the bisimulation game, names can be identified by a substitution σ. Thus, $[a = b]\tau$ and $\mathbf{0}$ are not considered bisimilar anymore, because, $\sigma([a = b]\tau)$ perform a τ transition if σ identifies a and b. Now consider $\nu a.[a = b]\tau$. It will never perform a τ transition, because a is restricted and then it cannot be identified with b. Thus in the bisimulation game, we have to avoid those substitutions that identify a and b. In order to properly handle the restriction operator, we have to introduce *distinctions*, i.e. relations that express permanent inequalities on names.

Definition 1 (Distinction). *A* distinction *D is a finite symmetric and irreflexive relation on names. A substitution σ* respects *D iff aDb implies $\sigma(a) \neq \sigma(b)$.*

In the following we will use $\mathcal{D}$ to mean the set of all distinctions and $\sigma(D)$ to mean the distinction $\{(\sigma(a), \sigma(b)) \mid (a, b) \in D\}$. Sometimes, in the expressions defining distinctions we shall avoid to give all the symmetric pairs; for instance, we might define $D = \{(a, b)\}$ without recalling that also $(b, a) \in D$. In the following definitions, a name declared *fresh* is supposed to be different from all the others appearing in the definition.

Definition 2. *Let $R = \{R_D \mid D \in \mathcal{D}\}$ be a $\mathcal{D}$ sorted family of symmetric relations. R is an* open bisimulation *iff $\forall D \in \mathcal{D}$ and $\forall \sigma$ respecting D, whenever pR_Dq:*

- *if $\sigma(p) \xrightarrow{\alpha} p'$ with $\mathrm{bn}(\alpha)$ fresh, then $\sigma(q) \xrightarrow{\alpha} q'$ and $p'R_{\sigma(D)}q'$,*
- *if $\sigma(p) \xrightarrow{\overline{a}(b)} p'$ with b fresh, then $\sigma(q) \xrightarrow{\overline{a}(b)}$ and $p'R_{D^*}q'$*

where $D^* = \sigma(D) \cup \{(b, i),\ \forall i \in \mathrm{fn}(\sigma(p) \cup \sigma(q))\}$.

We write $p \sim^O_D q$, if there is an open bisimulation R such that pR_Dq.

The intuitive meaning of the last clause, is that b is different from all the other free names appearing in $\sigma(p)$ and $\sigma(q)$ since it has been generated by some restriction νb. Thus we have to check that the arriving states p' and q' are bisimilar when considering b distinct form all the other names.

The definition of $\sim^O$ involves at each step a quantification over all substitutions. In [21], the author introduces a more efficient characterization of $\sim^O$, by defining a symbolic transition system. Labels on this LTS are pairs (M, μ) where M is a *match sequence* and μ is an action. A match sequence (ranged over by M, N) is a sequence of equalities of names of the form $[a = b]$. We will write MN to denote the concatenation of M and N and $M \rhd N$ if M implies N, i.e., whenever M holds, also N holds. Every matching sequence M defines an equivalence relation E_M. We denote by σ^M a special substitution that chooses a representative for each equivalence class of E_M, and maps every name in the representative of its class. Note that there may exists more than one σ^M, we just choose one of them.

The symbolic transition system is presented in Table 2. In the transition $p \xrightarrow{M,\mu}_e p'$, M represents the minimal substitution σ^M that allows p to perform

Table 2. Symbolic transition system for open π-calculus

$$(\textsc{pre})\ \alpha.p \xrightarrow{\varnothing,\alpha}_e p \qquad (\textsc{cls})\ \frac{p \xrightarrow{M,\overline{a}(x)}_e p' \quad q \xrightarrow{N,b(x)}_e q'}{p \mid q \xrightarrow{MN[a=b],\tau}_e \nu x.p' \mid q'} \qquad (\textsc{sum})\ \frac{p \xrightarrow{M,\mu}_e p'}{p+q \xrightarrow{M,\mu}_e p'}$$

$$(\textsc{par})\ \frac{p \xrightarrow{M,\mu}_e p'}{p \mid q \xrightarrow{M,\mu}_e p' \mid q}\ \operatorname{bn}(\mu) \cap \operatorname{fn}(q) = \varnothing \qquad (\textsc{com})\ \frac{p \xrightarrow{M,\overline{a}b}_e p' \quad q \xrightarrow{N,c(d)}_e q'}{p \mid q \xrightarrow{MN[a=c],\tau}_e p' \mid q'\{^b/_d\}}$$

$$(\textsc{rep})\ \frac{p \mid p! \xrightarrow{M,\mu}_e q}{p! \xrightarrow{M,\mu}_e q} \qquad (\textsc{mat})\ \frac{p \xrightarrow{M,\mu}_e p'}{[a=b]p \xrightarrow{M[a=b],\mu}_e p'}$$

$$(\textsc{res})\ \frac{p \xrightarrow{M,\mu}_e p'}{\nu b.p \xrightarrow{M,\mu}_e \nu b.p'}\ b \notin \operatorname{nm}(M \cup \mu) \qquad (\textsc{opn})\ \frac{p \xrightarrow{M,\overline{a}b}_e p'}{\nu b.p \xrightarrow{M,\overline{a}(b)}_e p'}\ b \notin \operatorname{nm}(m) \cup \{a\}$$

μ. As an example consider the processes $p = [a=b]\tau$ and $q = p + [c=d][a=b]\tau$. The process q can perform the transitions $q \xrightarrow{[a=b],\tau}_e \mathbf{0}$ and $q \xrightarrow{[a=b][c=d],\tau}_e \mathbf{0}$, while p performs only the former transition. However $p \sim^O q$.

Definition 3. *Let $R = \{R_D \mid D \in \mathcal{D}\}$ be a $\mathcal{D}$ sorted family of symmetric relations. R is an* efficient open bisimulation *iff $\forall D \in \mathcal{D}$, whenever pR_Dq:*

- *if $p \xrightarrow{M,\alpha}_e p'$ with* $\operatorname{bn}(\alpha)$ *fresh and M respects D, then $q \xrightarrow{N,\alpha'}_e q'$ such that $M \rhd N$, $\sigma^M(\alpha) \equiv_\alpha \sigma^M(\alpha')$ and $\sigma^M(p')R_{\sigma^M(D)}\sigma^M(q')$,*
- *if $p \xrightarrow{M,\overline{a}(b)}_e p'$ with b fresh and M respects D, then $q \xrightarrow{N,\overline{c}(b)}_e q'$ such that $M \rhd N$, $\sigma^M(\overline{a}(b)) \equiv_\alpha \sigma^M(\overline{c}(b))$ and $\sigma^M(p')R_{D^*_M}\sigma^M(q')$*

where $D^*_M = \sigma^M(D) \cup \{(b,i),\ \ \forall i \in \operatorname{fn}(\sigma^M(p) \cup \sigma^M(q))\}$.

We write $p \asymp_D q$, if there is an efficient open bisimulation R such that pR_Dq.

Intuitively the above clauses ensure that in the ordinary transition system, the move $\sigma^M(p) \xrightarrow{\sigma^M(\alpha)} \sigma^M(p')$ is matched by $\sigma^M(q) \xrightarrow{\sigma^M(\alpha')} \sigma^M(q')$. In [21], it is proved that $\asymp$ and $\sim^O$ coincide, but the former is more efficient than the latter, since $\asymp$ forces only those fusions of names which are strictly necessary to ensure the equivalence, while $\sim^O$ forces all the fusions.

The asynchronous fragment. The asynchronous π-calculus [9,1] is defined as a subset of π, without output prefixes and outputs in choice points. Formally:

$$p ::= \overline{a}b,\ \ p_1 \mid p_2,\ \ \nu a.p,\ \ !g,\ \ g \qquad g ::= \mathbf{0},\ \ \alpha.p,\ \ g_1 + g_2 \qquad \alpha ::= a(b),\ \ \tau$$

The operational semantics is obtained by replacing the rules (sum) and (rep) of Table 1 with the three rules of Table 3. The main difference with standard π is in the notion of observation. Indeed in the asynchronous case, input are not observable. Intuitively an observer that sends a message to a system, cannot know if the system has received it. Thus bisimulation ignores input transitions.

Table 3. Operational semantics of the asynchronous π-calculus

$$(\textsc{out})\ \overline{a}b \xrightarrow{\overline{a}b} \mathbf{0} \qquad (\textsc{sum})\ \frac{g \xrightarrow{\mu} p}{g + g' \xrightarrow{\mu} p'} \qquad (\textsc{rep})\ \frac{g \xrightarrow{\mu} p'}{!g \xrightarrow{\mu} p' | !g}$$

Definition 4 ($o\tau$-bisimilarity). *A symmetric relation R is an $o\tau$-bisimulation iff, whenever pRq, if $p \xrightarrow{\mu} p'$ where μ is not an input action and* $\mathrm{bn}(\alpha)$ *is fresh, then $q \xrightarrow{\mu} q'$ and $p'Rq'$. Let $\sim^{o\tau}$ be the largest $o\tau$-bisimulation.*

Note that $a(x).\bar{c}x \sim^{o\tau} a(x).\overline{d}x$, even if the two processes are really different when they are put in parallel with a process $\overline{a}b$. In order to obtain an abstract semantics preserved under parallel composition, we proceed analogously to open bisimilarity, i.e. at any step of the bisimulation we put the process in parallel with all possible outputs (in the open we apply all possible substitutions).

Definition 5 (1-bisimilarity). *An 1-bisimulation is an $o\tau$-bisimulation R such that pRq implies $\forall \overline{a}b$, $(\overline{a}b \mid p)\, R\, (\overline{a}b \mid q)$. Let $\sim^1$ be the largest 1-bisimulation.*

Definition 6 (asynchronous bisimilarity). *A symmetric relation R is an asynchronous bisimulation iff it is an $o\tau$-bisimulation and whenever pRq, if $p \xrightarrow{a(b)} p'$, then either $q \xrightarrow{a(b)} q'$ and $p'Rq'$, or $q \xrightarrow{\tau} q'$ and $p'R(q' \mid \overline{a}b)$. Let $\sim^a$ be the largest asynchronous bisimulation*[1]*.*

In [1], it is proved that $\sim^1=\sim^a$. Our theory will formally show that $\sim^1=\sim^a$ and $\sim^O=\asymp$ are two instances of the same general concept. The abstract semantics $\sim^1$ and $\sim^O$ are saturated, i.e., obtained by closing w.r.t. all contexts, while $\sim^a$ and $\asymp$ are symbolic, i.e., efficient characterizations of the saturated ones.

3 Saturated and Symbolic Semantics

A *closed many-sorted unary signature* (S, Σ) consists of a set of sorts S, and an $S \times S$ sorted family $\Sigma = \{\Sigma_{s,t} \mid s, t \in S\}$ of sets of operation symbols which are closed under composition, that is if $f \in \Sigma_{s,t}$ and $g \in \Sigma_{t,u}$, then $g \circ f \in \Sigma_{s,u}$. Given $f \in \Sigma_{u,v}, g \in \Sigma_{t,u}, h \in \Sigma_{s,t}$, $f \circ (g \circ h) = (f \circ g) \circ h$ and moreover $\forall s \in S$, $\exists id_s \in \Sigma_{s,s}$ such that $\forall f \in \Sigma_{s,t}$, $id_t \circ f = f$ and $f \circ id_s = f$. A (S, Σ)-algebra $\mathbb{A}$ consists of an S sorted family $|\mathbb{A}| = \{A_s \mid s \in S\}$ of sets and a function $f_{\mathbb{A}} : A_s \rightarrow A_t$ for all $f \in \Sigma_{s,t}$ such that $(g \circ f)_{\mathbb{A}} = g_{\mathbb{A}}(f_{\mathbb{A}}(-))$ and $id_{s\mathbb{A}}$ is the identity function on A_s[2]. When $\mathbb{A}$ is clear from the context, we will write f to mean $f_{\mathbb{A}}$, and we will write A_s to mean the set of sort s of the family $|\mathbb{A}|$. Given $f \in \Sigma_{s,t}$, we call s the *source* of f, and t the *target*.

[1] In [1], $\sim^a$ was originally defined in the early LTS. The above definition, however coincides with the original $\sim^a$ because $\sim^a=\sim^g$ (Corollary 1, [1]).

[2] A closed many-sorted unary signature (S, Σ) is a category $\mathbf{C}$ and a (S, Σ)-algebra is a presheaf on $\mathbf{C}$. We adopt the above notation to be accessible to a wider audience.

In order to develop a general theory of bisimulation, we introduce *context interactive systems*. In our theory, an interactive system is a state-machine that can interact with the environment (contexts) through an evolving interface.

Definition 7 (Context Interactive System). *A* context interactive system *$\mathcal{I}$ is a quadruple $\langle(S,\Sigma),\mathbb{A},O,tr\rangle$ where:*

- *(S,Σ) is a closed many-sorted unary signature,*
- *$\mathbb{A}$ is a (S,Σ)-algebra,*
- *O is a set of observations,*
- *$tr \subseteq |\mathbb{A}|\times O\times|\mathbb{A}|$ is a labeled transition relation ($p \xrightarrow{o} p'$ means $(p,o,p')\in tr$).*

Roughly speaking sorts are interfaces of the system, while operators of Σ are contexts. Every state p with interface s (i.e. $p \in A_s$) can be inserted into the context $c \in \Sigma_{s,t}$, obtaining $c_{\mathbb{A}}(p)$ that has interface t. Every state can evolve into a new state (possibly with different interface) producing an observation $o \in O$.

The abstract semantics of interactive systems is usually defined through behavioural equivalences. In this paper we propose a general notion of bisimilarity. The idea is that two states of a system are equivalent if they are indistinguishable from an external observer that, in any moment of their execution, can insert them into some environment and then observe some transition.

Definition 8 (Saturated Bisimilarity). *Let $\mathcal{I}=\langle(S,\Sigma),\mathbb{A},O,tr\rangle$ be a context interactive system. Let $R=\{R_s\subseteq A_s\times A_s \mid s\in S\}$ be an S sorted family of symmetric relations. R is a* saturated bisimulation *iff, $\forall s,t\in S$, $\forall c\in\Sigma_{s,t}$, whenever pR_sq, if $c_{\mathbb{A}}(p)\xrightarrow{o}p'$, then $c_{\mathbb{A}}(q)\xrightarrow{o}q'$ and $p'Rq'$.*

We write $p\sim_s^S q$ iff there is a saturated bisimulation R such that pR_sq.

An alternative but equivalent definition can be given by defining the *saturated transition system* (SATTS) as follows: $p \xrightarrow{c,o}_S q$ if and only if $c(p)\xrightarrow{o} q$. Trivially the standard bisimilarity over SATTS coincides with $\sim^S$.

Proposition 1. *$\sim^S$ is the coarsest bisimulation congruence.*

Saturated bisimulation is a good notion of equivalence but it is hard to check, since it involves a quantification over all contexts. A solution out of the impasse is suggested by $\asymp$. We can define a symbolic transition system where transitions are labeled both with the usual observation and also with the minimal context (substitutions, in the case of open π) that allows the transition.

Definition 9 (Symbolic Context Transition System). *A* symbolic context transition system *(SCTS for short) for a system $\mathcal{I}=\langle(S,\Sigma),\mathbb{A},O,tr\rangle$ is a transition system $\beta\subseteq|\mathbb{A}|\times\Sigma\times O\times|\mathbb{A}|$.*

In SCTS, we have two labels with different tastes. The first label is a context that tells us when the transition can be performed. We call this label the *interaction*, while the second is the *observation* produced by the transition. It is worth to note that in some formalisms interactions and observations coincide and thus only one label is necessary. However, in the general case, the two concepts are distinct as it is the case of asynchronous formalisms, where the input interaction cannot be observed. In the asynchronous π, the transition $a(x) \xrightarrow{a(x)} \mathbf{0}$, can be seen as $a(x) \xrightarrow{-|\overline{a}b,\tau}_{\beta} \mathbf{0}$, where $- \mid \overline{a}b$ is the interaction (i.e., the minimal context that allows the transition) and τ is the observation.

The intuitive meaning of such a transition is that for all larger contexts $- \mid \overline{a}b \mid Q$, we have that $a(x) \mid \overline{a}b \mid Q \xrightarrow{\tau} \mathbf{0} \mid Q$. The same happens in the symbolic LTS of $\sim^{O}$, where the transition $[a=b]\overline{c}d \xrightarrow{[a=b],\overline{c}d}_{e} \mathbf{0}$ roughly means that for all substitution σ that equate a and b, $\sigma([a=b]\overline{c}d) \xrightarrow{\sigma(\overline{c}d)} \sigma(\mathbf{0})$.

It is worth to note that there is a difference between the asynchronous and the open case. In the former, the insertion inside the context does not modify the observation, while in the open, the substitution is applied also to the observation. In general, contexts can modify in many different ways the execution of a transition. For example in the π-calculus, the prefix contexts $\alpha.-$ inhibits all the transitions (i.e. if $p \xrightarrow{\mu} p'$, then $\alpha.p \not\xrightarrow{\mu}$), while the context $\nu a.-$ inhibits only the transitions with subject a. For this reason we need a set of rules that formally describes how contexts modify transitions. Hereafter we fix the following format.

$$\frac{p_s \xrightarrow{o} q_t}{c(p_s) \xrightarrow{o'} d(q_t)}$$

This rule states that all processes with sort s that perform a transition with observation o going into q_t, when inserted into the context $c \in \Sigma_{s,s'}$ can perform a transition with the observation o' going into $d(q_t)$ for some context $d \in \Sigma_{t,t'}$.

In the following, we write $c \;\frac{o}{o'}\; d$ to mean a rule like the above. The rules $c \;\frac{o}{o'}\; c'$ and $d \;\frac{o'}{o''}\; d'$ *derive* the rule $d \circ c \;\frac{o}{o''}\; d' \circ c'$ if $d \circ c$ and $d' \circ c'$ are defined. Given a set of rules $\mathcal{R}$, $\Phi(\mathcal{R})$ is the set of all the rules derivable from $\mathcal{R}$ together with the identities rules ($\forall o \in O$ and $\forall s,t \in S$, $id_s \;\frac{o}{o}\; id_t$).

Definition 10 (Saturation). *Let $\mathcal{I} = \langle (S,\Sigma), \mathbb{A}, O, tr \rangle$ be a context interactive system, β an* SCTS *and $\mathcal{R}$ a set of rules of the format described above. The* saturation *of β w.r.t. $\mathcal{R}$ (denoted by $\mathcal{R}(\beta)$) is the transition system described as:*

$$\frac{p \xrightarrow{c,o}_{\beta} p' \quad d \;\frac{o}{o'}\; e \in \Phi(\mathcal{R})}{p \xrightarrow{d \circ c, o'}_{\mathcal{R}(\beta)} e(p')}.$$

We say that β and $\mathcal{R}$ are sound and complete *w.r.t. $\mathcal{I}$ if the saturation of β w.r.t. $\mathcal{R}$ coincides with* SATTS*, i.e., $p \xrightarrow{c,o}_{\mathcal{R}(\beta)} p'$ iff $p \xrightarrow{c,o}_{S} p'$ (i.e., iff $c(p) \xrightarrow{o} p'$). A transition $p \xrightarrow{c,o} q$* dominates *$p \xrightarrow{c',o'} q'$ if there exists $d \;\frac{o}{o'}\; e \in \Phi(\mathcal{R})$ such that $c' = d \circ c$ and $q' = q(e)$.*

A sound and complete SCTS could be considerably smaller than the saturated transition system, but still containing all the information needed to recover $\sim^S$. Note that the standard bisimilarity over SCTS is usually stricter than $\sim^S$. Consider a process p that performs only the transitions $p \xrightarrow{c,o}_\beta p_1$ and $p \xrightarrow{c',o'}_\beta p_2$ such that $c' = d \circ c$, $d \;\frac{\quad o \quad}{o'}\; e$, $e(p_1) \sim^S p_2$. Now take a process q that performs only $q \xrightarrow{c,o}_\beta q_1$ such that $p_1 \sim^S q_1$. Clearly p and q are not bisimilar on SCTS, because $p \xrightarrow{c',o'}_\beta p'$ while q cannot. However $p \sim^S q$, because $q \xrightarrow{c',o'}_{\mathcal{R}(\beta)} e(q_1)$ (i.e., $c'(q) \xrightarrow{o'} e(q_1)$) and, since $q_1 \sim^S p_1$, then $e(q_1) \sim^S e(p_1) \sim^S p_2$.

Definition 11 (Symbolic and Semi-Saturated Bisimilarity). *Let $\mathcal{I} = \langle (S, \Sigma), \mathbb{A}, O, tr \rangle$ be an interactive system, $\mathcal{R}$ be a set of rules and β be a symbolic transition system. Let $R = \{R_s \subseteq A_s \times A_s \mid s \in S\}$ be an S sorted family of symmetric relations. R is a* symbolic bisimulation *iff $\forall s \in S$, whenever pR_sq:*

- *if $p \xrightarrow{c',o'}_\beta p'_1$, then $\exists d \;\frac{\quad o \quad}{o'}\; d' \in \Phi(\mathcal{R})$, $d \circ c = c'$, $q \xrightarrow{c,o}_\beta q_1$ and $p'_1 R d'(q_1)$.*

We write $p \sim_s^{SYM} q$ iff there exists a symbolic bisimulation R such that pR_sq. Semi-saturated bisimilarity *($\sim^{SS}$) is obtained replacing the above condition with:*

- *if $p \xrightarrow{c',o'}_\beta p'_1$, then $c'(q) \xrightarrow{o'} q'_1$ and $p'_1 R q'_1$.*

Theorem 1. *Let $\mathcal{I}$ be a context interactive system, β an* SCTS *and $\mathcal{R}$ a set of rules. If β and $\mathcal{R}$ are sound and complete w.r.t. $\mathcal{I}$, then $\sim^{SYM} = \sim^{SS} = \sim^S$.*

4 Context Interactive Systems for π-Calculus

In this section we present context interactive systems for asynchronous (4.1) and open (4.2) π-calculus. In the former, contexts are parallel output processes, saturated bisimilarity coincides with $\sim^1$, while symbolic bisimilarity coincides with $\sim^a$. In the latter, contexts are fusions of names that respect distinctions, saturated bisimilarity coincides with $\sim^O$ and symbolic bisimilarity with $\asymp$.

4.1 Asynchronous

We assume the set of names $\mathcal{N}$ to be totally ordered . With n we mean both the nth names and the set of names smaller or equal than n. The context interactive system for asynchronous π-calculus is $\mathcal{A} = \langle (S^{\mathcal{A}}, \Sigma^{\mathcal{A}}), \mathbb{A}, O_{\mathcal{A}}, tr_{\mathcal{A}} \rangle$.

The many-sorted signature $(S^{\mathcal{A}}, \Sigma^{\mathcal{A}})$ is formally defined as

- $S^{\mathcal{A}} = \omega$ (the set of natural numbers);
- $\Sigma^{\mathcal{A}}_{n,m}$ with $m \geq n$ is the set of contexts c generated by $c ::= -,\ - \mid \overline{a}b$, where $a \in n$ and $b \in m$.[3]

[3] $\forall n \in \omega$, id_n is $- \in \Sigma^{\mathcal{A}}_{n,n}$, while $\circ$ is the syntactic composition of contexts.

Let us define the $(S^{\mathcal{A}}, \Sigma^{\mathcal{A}})$-algebra $\mathbb{A}$. For every sort n, A_n is the set of asynchronous π-processes p such that $n \geq max\ \mathrm{fn}(p)$. Then $\forall p \in A_n$ and $\forall c \in \Sigma^{\mathcal{A}}_{n,m}$, $c_{\mathbb{A}}(p)$ is the process of sort m obtained by syntactically inserting p into c. In this system, interfaces are sets of names. A process with interface n uses only names in n (not all, just a part) and can be put in parallel with outputs sending messages over n. Given a process p and a natural number $n \geq max\ \mathrm{fn}(p)$, we denote with p_n the process p with interface n.

The set of observations is $O_{\mathcal{A}} = \{\overline{a}b, \overline{a}(), \tau \mid a, b \in \mathcal{N}\}$. Note that the input action is not an observation, since in the asynchronous case it is not observable. Moreover note that in the bound output, the sent name does not appear. This is because, any process with sort n will send as bound output the name $n+1$.

The following rules define the transition structure $tr_{\mathcal{A}}$ (denoted by $\rightarrow_{\mathcal{A}}$).

$$\frac{p \xrightarrow{\overline{a}b} p'}{p_n \xrightarrow{\overline{a}b}_{\mathcal{A}} p'_n} \quad \frac{p \xrightarrow{\tau} p'}{p_n \xrightarrow{\tau}_{\mathcal{A}} p'_n} \quad \frac{p \xrightarrow{\overline{a}(n+1)} p'}{p_n \xrightarrow{\overline{a}()}_{\mathcal{A}} p'_{n+1}}$$

Proposition 2. *Let p, q be asynchronous π-processes, and let $n \geq max\ \mathrm{fn}(p \cup q)$. Then $p \sim^1 q$ iff $p_n \sim^S_n q_n$.*

The above result states that $\sim^1$ is an instance of the more general concept of saturated bisimilarity. In the rest of this subsection, we will show that $\sim^a$ is an instance of symbolic bisimilarity. The SCTS for the asynchronous π is defined by the following rule, where $- \in \Sigma^{\mathcal{A}}_{n,n}$ and $- \mid \overline{a}m \in \Sigma^{\mathcal{A}}_{n,x}$.

$$\frac{p \xrightarrow{\overline{a}b} p'}{p_n \xrightarrow{-,\overline{a}b}_{\alpha} p'_n} \quad \frac{p \xrightarrow{\overline{a}(n+1)} p'}{p_n \xrightarrow{-,\overline{a}()}_{\alpha} p'_{n+1}} \quad \frac{p \xrightarrow{\tau} p'}{p_n \xrightarrow{-,\tau}_{\alpha} p'_n} \quad \frac{p \xrightarrow{a(m)} p' \quad x = max\{m, n\}}{p_n \xrightarrow{-|\overline{a}m,\tau}_{\alpha} p'_x}$$

Note that the only non standard rule is the fourth. If, in the standard transition system a process can perform an input, in the SCTS the same process can perform a τ, provided that there is an output processes in parallel. Note that the sort of the arriving state depends on the name received m: if it is smaller than n, then the arriving sort is n, otherwise it is m.

Now we have to define a set of rules $\mathcal{R}_{\mathcal{A}}$ that describes how contexts transforms transitions. Since our contexts are just parallel outputs, all the contexts preserve transitions. This is expressed by the following (parametric) rules

$$\frac{P_n \xrightarrow{\tau}_{\mathcal{A}} P'_n}{c(P_n) \xrightarrow{\tau}_{\mathcal{A}} c(P'_n)} \quad \frac{P_n \xrightarrow{\overline{a}b}_{\mathcal{A}} P'_n}{c(P_n) \xrightarrow{\overline{a}b}_{\mathcal{A}} c(P'_n)} \quad \frac{P_n \xrightarrow{\overline{a}(b)}_{\mathcal{A}} P'_{n+1}}{c(P_n) \xrightarrow{\overline{a}(b)}_{\mathcal{A}} c^{+1}(P'_{n+1})}$$

where $c \in \Sigma^{\mathcal{A}}_{n,m}$ is a generic context, and $c^{+1} \in \Sigma^{\mathcal{A}}_{n+1,m+1}$ is the same syntactic context as c, but with different sorts. Instantiating the general definition of symbolic bisimulation to α and $\mathcal{R}_{\mathcal{A}}$, we retrieve the definition of asynchronous bisimulation. Indeed transitions of the form $p \xrightarrow{-,\mu}_{\alpha} p'$ (in the original LTS τ and output), can be matched only by transitions with the same label, while transitions $p \xrightarrow{-|\overline{a}m,\tau}_{\alpha} p'$ (the input) can be matched either by $q \xrightarrow{-|\overline{a}m,\tau}_{\alpha} q'$ using the rule $id \;\overset{\tau}{\underset{\tau}{\text{———}}}\; id$ or by $q \xrightarrow{-|,\tau}_{\alpha} q'$ using the rule $- \mid \overline{a}m \;\overset{\tau}{\underset{\tau}{\text{———}}}\; - \mid \overline{a}m$.

Proposition 3. *Let p, q be asynchronous π-processes, and let $n \geq max\ \mathrm{fn}(p \cup q)$. Then $p \sim^{a} q$ iff $p_n \sim_{n}^{SYM} q_n$.*

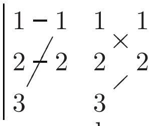
Proposition 4. *α and $\mathcal{R}_{\mathcal{A}}$ are sound and complete w.r.t. $\mathcal{A}$.*

Corollary 1 (by Thm. 1). *$\sim^{1} = \sim^{a}$ as shown in [1].*

4.2 Open

In this section we will present $\mathcal{O} = \langle (S^{\mathcal{O}}, \Sigma^{\mathcal{O}}), \mathbb{O}, O_{\mathcal{O}}, tr_{\mathcal{O}} \rangle$ for open π-calculus. As in the asynchronous case, we assume the set of names $\mathcal{N}$ to be totally ordered[4], and we write n to mean the set of names smaller or equal to n. A *fusion* $\sigma : n \to m$ is a surjective function where

$$\sigma(i) < \sigma(j) \Rightarrow \exists k \in \sigma^{-1}(\sigma(i)) \text{ such that } k < j.$$



The above condition guarantees that fusions are in one to one correspondence with the equivalence classes on names and thus with matching sequences. For example consider the two functions depicted above on the right. Both represents the matching $[1 = 3]$, but only the leftmost is a fusion.

The multi-sorted signature $(S^{\mathcal{O}}, \Sigma^{\mathcal{O}})$ is formally defined as

- $\Sigma^{\mathcal{O}} = \{(n, D)$ for $n \in \omega$ and $D \in \mathcal{D}$ such that $\mathrm{nm}(D) \subseteq n\}$;
- $\Sigma^{\mathcal{O}}_{(n,D),(n',D')}$ is the set of fusions $\sigma : n \to n'$ such that:
  1. Respect distinction, i.e., $iDj \Rightarrow \sigma(i) \neq \sigma(j)$,
  2. Preserve distinction, i.e., $iDj \Rightarrow \sigma(i)D'\sigma(j)$.[5]

Let us define the $(S^{\mathcal{O}}, \Sigma^{\mathcal{O}})$-algebra $\mathbb{O}$. For every sort (n, D), $O_{n,D}$ is the set of processes p such that $n \geq max\{\mathrm{fn}(p)\}$. Then $\forall p \in O_{n,D}$ and $\forall \sigma \in \Sigma^{\mathcal{O}}_{(n,D),(n',D')}$, $\sigma_{\mathbb{O}}(p)$ is the process of sort (n', D') obtained by replacing in p all the occurrences of $a \in \mathrm{fn}(p)$ with $\sigma(a)$. In this system, interfaces are pairs (n, D) where n is a set of names (as in the asynchronous case) and D is a distinction. A process with interface (n, D), can be inserted only in those fusions that respect D. Given a process p, a natural number $n \geq max\ \mathrm{fn}(p)$ and D such that $\mathrm{nm}(D) \subseteq n$, we denote with $p_{n,D}$ the process p with interface (n, D).

The set of observations is $O_{\mathcal{O}} = \{a(), \overline{a}b, \overline{a}(), \tau \mid a, b \in \mathcal{N}\}$. Differently from the asynchronous case, here input is observable. However note that the received name does not appear. This is because any process with sort (n, D) will receive the name $n + 1$ (that could be later fused with other names).

The following rules define the transition structure $tr_{\mathcal{O}}$ (denoted by $\to_{\mathcal{O}}$).

[4] We can work with not ordered $\mathcal{N}$ by taking as signature the category $\mathbf{D}$ of [13].

[5] $\forall (n, D) \in \Sigma^{\mathcal{O}}$, $id_{n,D}$ is the identity fusion, while $\circ$ is composition of substitutions.

$$\frac{p \xrightarrow{\overline{a}b} p'}{p_{n,D} \xrightarrow{\overline{a}b}_{\mathcal{O}} p'_{n,D}} \quad \frac{p \xrightarrow{\tau} p'}{p_{n,D} \xrightarrow{\tau}_{\mathcal{O}} p'_{n,D}} \quad \frac{p \xrightarrow{a(n+1)} p'}{p_{n,D} \xrightarrow{a()}_{\mathcal{O}} p'_{n+1,D}} \quad \frac{p \xrightarrow{\overline{a}(n+1)} p'}{p_{n,D} \xrightarrow{\overline{a}()}_{\mathcal{O}} p'_{n+1,\bar{D}}}$$

where $\bar{D} = D \cup \{(n+1, i), \quad \forall i < n+1\}$. The only non-standard transition is the bound output, where in the arriving state the distinction $\overline{D}$ is forced.

Proposition 5. *Let $p, q \in O$ be π-processes, and let $n \geq max$ $\mathrm{fn}(p \cup q)$ and $\mathrm{nm}(D) \subseteq n$. Then $p \sim^{O}_{D} q$ iff $p_{n,D} \sim^{S}_{n,D} q_{n,D}$.*

In order to define the symbolic transition system we have to fix some notations. For any $\sigma \in \Sigma^{\mathcal{O}}_{(n,D),(m,D')}$, we denote by $\sigma^{+1} \in \Sigma^{\mathcal{O}}_{(n+1,D),(m+1,D')}$ the fusion that maps $n+1$ into $m+1$ and all the $i \leq n$ into $\sigma(i)$, while we use $\overline{\sigma^{+1}} \in \Sigma^{\mathcal{O}}_{(n+1,\overline{D}),(m+1\overline{D'})}$ to mean σ^{+1} with the enforced distinction $\overline{D'}$. For any matching sequence M that respects D, we denote by $\sigma^{M} \in \Sigma^{\mathcal{O}}_{(n,D),(m,\sigma^{M}(D))}$ the unique fusion corresponding to M.

The following rules define the SCTS by relying on the symbolic transition system presented in Sec. 2. In all the rules, we implicitly assume as premise that σ^{M} respects D.

$$\frac{p \xrightarrow{M,\overline{a}b}_{e} p'}{p_{n,D} \xrightarrow{\sigma^{M},\overline{\sigma^{M}(a)}\sigma^{M}(b)}_{o} \sigma^{M}(p'_{n,D})} \qquad \frac{p \xrightarrow{M,\tau}_{e} p'}{p_{n,D} \xrightarrow{\sigma^{M},\tau}_{o} \sigma^{M}(p'_{n,D})}$$

$$\frac{p \xrightarrow{M,a(n+1)}_{e} p'}{p_{n,D} \xrightarrow{\sigma^{M},\sigma^{M}(a)()}_{o} \sigma^{M^{+1}}(p'_{n+1,D})} \quad \frac{p \xrightarrow{M,\overline{a}(n+1)}_{e} p'}{p_{n,D} \xrightarrow{\sigma^{M},\overline{\sigma^{M}(a)}()}_{o} \overline{\sigma^{M^{+1}}}(p'_{n+1,\overline{D}})}$$

Our SCTS differs form the canonical symbolic transition system, because the substitution here is applied both to observations and arriving states. Now we have to fix a set of rules $\mathcal{R}_{\mathcal{O}}$ that describes how fusions transform transitions. It is well known from [16] that substitutions preserve all the transitions by applying the substitution also to the observation. This is expressed by the following parametric rules for every $\sigma \in \Sigma^{\mathcal{O}}_{(n,D),(n',D')}$.

$$\frac{P_{n,D} \xrightarrow{\tau} P'_{n,D}}{\sigma(P_{n,D}) \xrightarrow{\tau} \sigma(P'_{n,D})} \qquad \frac{P_{n,D} \xrightarrow{\overline{a}b} P'_{n,D}}{\sigma(P_{n,D}) \xrightarrow{\overline{\sigma(a)}\sigma(b)} \sigma(P'_{n,D})}$$

$$\frac{P_{n,D} \xrightarrow{a()} P'_{n+1,D}}{\sigma(P_{n,D}) \xrightarrow{\sigma(a)()} \sigma^{+1}(P'_{n+1,D})} \quad \frac{P_{n,D} \xrightarrow{\overline{a}()} P'_{n+1,\overline{D}}}{\sigma(P_{n,D}) \xrightarrow{\overline{\sigma(a)}()} \overline{\sigma^{+1}}(P'_{n+1,\overline{D}})}$$

Proposition 6. *Let $p, q \in O$ be π-processes, and let $n \geq max$ $\mathrm{fn}(p \cup q)$ and $\mathrm{nm}(D) \subseteq n$. Then $p \asymp_{D} q$ iff $p_{n,D} \sim^{SYM}_{n,D} q_{n,D}$.*

Proposition 7. *o and $\mathcal{R}_{\mathcal{O}}$ are sound and complete w.r.t. $\mathcal{O}$.*

Corollary 2 (by Thm. 1). *$\sim^{O} = \asymp$ as shown in [21].*

5 Open Petri Nets

Differently from process calculi, Petri nets have not a widely known interactive behaviour. Indeed they model concurrent systems that are closed, in the sense that they do not interact with the environment. *Open nets* [10,2] are P/T Petri nets that can interact by exchanging tokens on *input* and *output places.*

Given a set X, we write $X^{\oplus}$ for the free commutative monoid over X. A multiset $m \in X^{\oplus}$ is a function from X to ω (the set of natural number) that associates a multiplicity to every element of X. Given two multisets m_1 and m_2, $m_1 \oplus m_2$ is defined as $\forall x \in X$, $m_1 \oplus m_2(x) = m_1(x) + m_2(x)$. We write $m_1 \subseteq m_2$ if $\forall x \in X$, $m_1(x) \leq m_2(x)$. If $m_1 \subseteq m_2$, the multiset $m_2 \ominus m_1$ is defined as $\forall x \in X$ $m_2 \ominus m_1(x) = m_2(x) - m_1(x)$. Given a set $Y \subseteq X$, and $m \in X^{\oplus}$, the multiset $m \upharpoonright Y$ is defined as $m \upharpoonright Y(x) = m(x)$ if $x \in Y$, 0 otherwise. We write $\varnothing$ to denote both the empty set and the empty multiset.

Definition 12 (open net). *An open net is a tuple $N = (S, T, pre, post, \lambda, I, O)$ where S is the set of places, T is the set of transitions (with $S \cap T = \varnothing$), $pre, post : T \to S^{\oplus}$ are functions mapping each transition to its pre- and post-set, $\lambda : T \to \Lambda$ is a labeling function (Λ is a set of labels) and $I, O \subseteq S$ are the sets of input and output places (with $I \cap O = \varnothing$). A marked open net is pair $\langle N, m \rangle$ where N is an open net and $m \in S^{\oplus}$ is a marking.*

Fig. 1 shows two open nets where, as usual, circles represents places and rectangles transitions (labeled with α, β, χ). Arrows from places to transitions represent *pre*, while arrows from transitions to places represent *post*. Input places are denoted by ingoing edges, while output places are denoted by outgoing edges. Thus in N_1, x and y are output places, while z is the only input place. In N_2, it is the converse. The *parallel composition* of the two nets is defined by attaching them on their input and output places. As an example, we can compose N_1 and N_2 by attaching them through x, y and z.

The operational semantics of marked open nets is expressed by the rules on Table 4 where, in order to make lighter the notation, we use ${}^{\bullet}t$ and $t^{\bullet}$ to denote $pre(t)$ and $post(t)$ and we avoid to put brackets around the marked net $\langle N, m \rangle$. The rule (TR) is the standard rule of P/T nets (seen as multisets rewriting), while the other two are specific of open nets. The rule (IN) states that in any moment a token can be inserted inside an input place and, for this reason, the LTS has always an infinite number of states. The rule (OUT) states that when a token is in an output place, it can be removed. Fig.1[A] shows part of the infinite transition system of $\langle N_2, a \rangle$.

The abstract semantics is defined in [3] as the standard bisimilarity (denoted by $\sim^N$) and it is a congruence under the parallel composition outlined above. This is due to the rules (IN) and (OUT), since they put a marked net in all the possible contexts. If we consider just the rule (TR), then bisimilarity fails to be a congruence. Thus also for open nets, the canonical definition of bisimulation consists of inserting the system in all the possible contexts and observing what

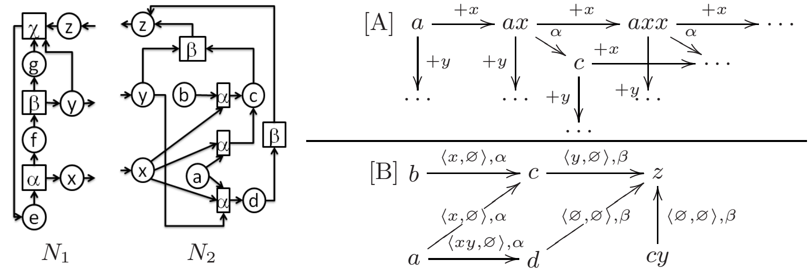


Fig. 1. N_1 and N_2 are two open Petri nets. [A] Part of the infinite transition system of $\langle N_2, a\rangle$. [B] The symbolic transition system of $\langle N_2, a\rangle$, $\langle N_2, b\rangle$ and $\langle N_2, cy\rangle$.

happens, but differently from open and asynchronous bisimilarity, a symbolic LTS and an efficient characterization of $\sim^N$ has never been given.

5.1 Context Interactive System for Open Nets

In this section we introduce the context interactive system for open nets $\mathcal{N} = \langle (S^{\mathcal{N}}, \Sigma^{\mathcal{N}}), \mathbb{N}, \Lambda, tr_{\mathcal{N}}\rangle$. Contexts are insertions and the deletions of tokens.

The many-sorted signature $(S^{\mathcal{N}}, \Sigma^{\mathcal{N}})$ is formally defined as:

- $S^{\mathcal{N}} = \{(I, O, m) \mid\ m \in O^{\oplus}\}$,
- $\Sigma^{\mathcal{N}}_{(I,O,m),\langle I,O,m'\rangle} = \{\langle i, o\rangle \mid\ i \in I^{\oplus},\ o \in O^{\oplus},\ o \subseteq m,\ m' = m \ominus o\}$.[6]

The sorts of the signature are triples (I, O, m) where I and O are sets of input places and output places and $m \in O^{\oplus}$ is a marking on the output places.

Operators of $\Sigma^{\mathcal{N}}$ are pairs $\langle i, o\rangle \in \Sigma^{\mathcal{N}}_{(I,O,m),\langle I,O,m'\rangle}$ where $i \in I^{\oplus}$, $o \in O^{\oplus}$ are, respectively, multisets of tokens added in the input places and removed from the output places. Note that in the target sort, the set of input and output places are the same of the source (meaning that context cannot modify I and O), while the marking $m' \in O^{\oplus}$ is equal to $m \ominus o$.

We say that an open net N has interface (I, O) if I and O are respectively its sets of input and output places. While a marked open net $\langle N, m\rangle$ has interface (I, O, m') if (I, O) is the interface of N and moreover if $m' = m \upharpoonright O$. This means that tokens in the output places are visible from the environment, while tokens in the input places are not. We can better understand this difference, by observing that the environment can remove tokens in the output places only if they are present, while it can always add tokens in the input places.

Let us define the $(S^{\mathcal{N}}, \Sigma^{\mathcal{N}})$-algebra $\mathbb{N}$. For any sort (I, O, m), the carrier set $N_{I,O,m}$ contains all the marked open nets with interface (I, O, m). Any operator $\langle i, o\rangle \in \Sigma_{(I,O,m),(I,O,m')}$ is defined as the function that maps $\langle N, m_1\rangle$ into $\langle N, m_1 \oplus i \ominus o\rangle$. The transition structure $tr_{\mathcal{N}}$ (denoted by $\rightarrow_{\mathcal{N}}$) associates to a state $\langle N, m\rangle$ the transitions obtained by using the rule (TR) of Table 4.

[6] $\forall (I, O, m) \in S^{\mathcal{N}}$, $id_{I,O,m}$ is $\langle \varnothing, \varnothing\rangle$, while $\langle i_1, o_1\rangle \circ \langle i_2, o_2\rangle = \langle i_1 \oplus i_2, o_1 \oplus o_2\rangle$.

Table 4. Operational Semantics of marked open nets

$$(\textsc{tr})\ \frac{t \in T \quad \lambda(t) = l \quad m = {}^{\bullet}t \oplus c}{N, m \xrightarrow{l} N, t^{\bullet} \oplus c} \quad (\textsc{in})\ \frac{i \in I_N}{N, m \xrightarrow{+i} N, m \oplus i} \quad (\textsc{out})\ \frac{o \in O_N \quad o \in m}{N, m \xrightarrow{-o} N, m \ominus o}$$

Proposition 8. *Let $\langle N_1, m_1 \rangle$ and $\langle N_2, m_2 \rangle$ be two marked nets both with interface (I, O, m). Thus $\langle N_1, m_1 \rangle \sim^N \langle N_2, m_2 \rangle$ iff $\langle N_1, m_1 \rangle \sim^S_{I,O,m} \langle N_2, m_2 \rangle$.*

5.2 A Symbolic Semantics for Open Nets

In the case of open and asynchronous π-calculus, we already knew the symbolic transition system by classical results in literature. In the case of open nets, no symbolic semantics does exists, and thus we have to define it. We use exactly the same intuition underlying the symbolic LTS of open and asynchronous, i.e., we consider the *minimal contexts* that allow a given system to perform a transition.

The SCTS for open nets, η is defined by the following rule.

$$\frac{t \in T \quad \lambda(t) = l \quad m = (m \cap {}^{\bullet}t) \oplus c \quad i \subseteq I^{\oplus} \quad {}^{\bullet}t = (m \cap {}^{\bullet}t) \oplus i \quad o \subseteq c \upharpoonright O}{N, m \xrightarrow{\langle i, o \rangle, l}_{\eta} N, t^{\bullet} \oplus c \ominus o}$$

The marking $m \cap {}^{\bullet}t$ contains all the tokens of m that are needed to perform t. The marking c contains all the tokens of m that are not useful for performing t, while the marking i contains all the tokens that m needs to reach ${}^{\bullet}t$. Note that i is exactly the *smallest* multiset that is needed to perform the transition t. Indeed if we take i_1 strictly included into i, $m \oplus i_1$ cannot match ${}^{\bullet}t$.

As an example consider the net N_1 in Fig. 1 with marking gxy and let t be the only transition labeled with χ. We have that $gxy \cap {}^{\bullet}t = gy$, $c = x$ and $i = z$. Thus $N_1, gxy \xrightarrow{\langle z, x \rangle, \chi}_{\eta} N_1, e$ and also $N_1, gxy \xrightarrow{\langle z, \varnothing \rangle, \chi}_{\eta} N_1, ex$. In the former transition we have taken o equal to $x = c \upharpoonright O$, while in the latter $o = \varnothing$. The multiset $c \upharpoonright O$ is the largest that can be safely removed by m without inhibiting the transition t. Differently than input, in the output we have to consider both the transitions (expressed by the premise $o \subseteq c \upharpoonright O$) because one cannot *dominate* (in the sense of Def. 10) the other. Indeed the former cannot dominate the latter because there are no contexts that add tokens in the output places, while the latter cannot dominate the former because in general, we cannot know if removing tokens from output places preserves a transition.

This is expressed by the set of rules $\mathcal{R}_{\mathcal{N}}$ that is defined by the following parametric rule.

$$\frac{N, m \xrightarrow{l}_{\mathcal{N}} N, m'}{\langle i, \varnothing \rangle (N, m) \xrightarrow{l}_{\mathcal{N}} \langle i, \varnothing \rangle (N, m')}$$

This rule states that the addition of tokens in the input places preserves transitions. While it does not state anything about the deletion of tokens. Indeed an output place could be in the precondition of some transition (e.g., y in the net

N_1 in Fig. 1) and thus, the deletion of some tokens can inhibit the transition. Fig. 1[B] shows the SCTS of $\langle N_2, a\rangle$ and $\langle N_2, b\rangle$. The former perform a transition with $\langle xy, \varnothing\rangle$, while the latter cannot. However they are saturated bisimilar.

Proposition 9. *η and $\mathcal{R}_{\mathcal{N}}$ are sound and complete w.r.t. $\mathcal{N}$.*

The above proposition together with Thm. 1 state that symbolic and semi-saturated bisimilarity coincide with $\sim^S$. In the following we instantiate their general definition to $\mathcal{N}$ and $\mathcal{R}_{\mathcal{N}}$.

Definition 13 (Symbolic and semi-saturated bisimulation for nets). *Let $R = \{R_{I,O,m} \subseteq N_{I,O,m} \times N_{I,O,m} \mid (I,O,m) \in S^{\mathcal{N}}\}$ be a $S^{\mathcal{N}}$ sorted family of symmetric relations. R is a* symbolic bisimulation *iff $\forall (I,O,m) \in S^{\mathcal{N}}$, whenever $\langle N_1, m_1\rangle R_{I,O,m} \langle N_2, m_2\rangle$*

- *if $\langle N_1, m_1\rangle \xrightarrow{\langle i,o\rangle, l}_{\eta} \langle N_1, m_1'\rangle$ then $\exists i_1, x \in I^{\oplus}$ such that:*
  $i = i_1 \oplus x$, $\langle N_2, m_2\rangle \xrightarrow{<i_1,o>, l}_{\eta} \langle N_2, m_2'\rangle$ and $\langle N_1, m_1'\rangle R \langle N_2, m_2 \oplus x\rangle$.

R is a semi-saturated *bisimulation iff whenever $\langle N_1, m_1\rangle R_{I,O,m} \langle N_2, m_2\rangle$*

- *if $\langle N_1, m_1\rangle \xrightarrow{\langle i,o\rangle, l}_{\eta} \langle N_1, m_1'\rangle$ then $\langle N_2, m_2 \oplus i \ominus o\rangle \xrightarrow{l} \langle N_2, m_2'\rangle$ and $\langle N_1, m_1'\rangle R \langle N_2, m_2'\rangle$.*

6 Leifer and Milner Reactive Systems

As stated in the introduction, our approach generalizes the theory of reactive system by Leifer and Milner [11]. They define the syntax of the formalism through a (Lawvere-like) category **C** whose arrows are contexts and terms are arrows having as source a special object 0. In our theory, **C** is the closed many-sorted unary signature (S, Σ): objects are sorts and arrows are operators. Every term $p : 0 \to s$ is an element of the carrier-set A_s. Given a context $c : s \to t$, the composition $p; c$ is defined as $c_{\mathbb{A}}(p)$. They also define a subcategory **D** of reactive arrows. This is modeled in our formalism by adding for every arrow $d \in \mathbf{D}$ a rule as the following: $\dfrac{p \to q}{d(p) \to d(q)}$.

They define the reaction relation by closing some reaction rules under all reactive contexts. In the same way, we start with some labeled transitions (that generalize rewriting rules) and we generalizes w.r.t. all the rules as the above. Idem-PushOut (IPO) represents the minimal context that allows a reaction. The transition system labeled with IPOs (ITS) is an instance of our SCTS. Indeed the saturation of it, trough the above rules exactly coincides with SATTS as formally shown in [5]. However, IPO-bisimilarity (that is a congruence under restrictive condition) is stricter than $\sim^S$. In [5], we have provided a symbolic bisimilarity for ITS (Thm. 3) and proved that it coincides with $\sim^S$. This result is thus a special case of Thm. 1 presented here.

7 Conclusions

In this paper we have introduced saturated bisimilarity for context interactive systems and an efficient way to characterize it through symbolic bisimilarity. We have shown that our theory works for real formalism re-deriving well-known semantics, namely, the saturated and symbolic versions of asynchronous and open bisimilarities. Moreover we have applied our approach to open Petri nets with the result of a new (at our knowledge, the first) symbolic semantics that efficiently characterizes canonical bisimilarity. Leifer and Milner reactive systems have been applied to open Petri nets (without observations on transitions) in [15,22], but the derived LTS is infinite.

Our theory generalizes Leifer and Milner reactive systems by allowing observations. We think that observations are usually necessary, since one label cannot represent at the same time both interaction and observation.

As next step, we would like to give a coalgebraic semantics for symbolic bisimilarity by extending *normalized coalgebras* [6] and by exploiting the connections with coalgebras on presheafs [7]. The coalgebraic approach might yield a general minimization algorithm working directly on the symbolic transition systems in the style of [19].

References

1. Amadio, R.M., Castellani, I., Sangiorgi, D.: On bisimulations for the asynchronous π-calculus. In: Sassone, V., Montanari, U. (eds.) CONCUR 1996. LNCS, vol. 1119, pp. 147–162. Springer, Heidelberg (1996)
2. Baldan, P., Corradini, A., Ehrig, H., Heckel, R.: Compositional semantics for open Petri nets based on deterministic processes. M.S.C.S 15(1), 1–35 (2005)
3. Baldan, P., Corradini, A., Ehrig, H., Heckel, R., König, B.: Bisimilarity and behaviour-preserving reconfiguration of open petri nets. In: Mossakowski, T., Montanari, U., Haveraaen, M. (eds.) CALCO 2007. LNCS, vol. 4624, pp. 126–142. Springer, Heidelberg (2007)
4. Bonchi, F., Gadducci, F., König, B.: Process bisimulation via a graphical encoding. In: Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G. (eds.) ICGT 2006. LNCS, vol. 4178, pp. 168–183. Springer, Heidelberg (2006)
5. Bonchi, F., König, B., Montanari, U.: Saturated semantics for reactive systems. In: LICS, pp. 69–80. IEEE, Los Alamitos (2006)
6. Bonchi, F., Montanari, U.: Coalgebraic models for reactive systems. In: ECML 2007. LNCS, vol. 4701, pp. 364–380. Springer, Heidelberg (2007)
7. Fiore, M.P., Turi, D.: Semantics of name and value passing. In: LICS, pp. 93–104. IEEE, Los Alamitos (2001)
8. Hennessy, M., Lin, H.: Symbolic bisimulations. T.C.S. 138(2), 353–389 (1995)
9. Honda, K., Tokoro, M.: An object calculus for asynchronous communication. In: America, P. (ed.) ECOOP 1991. LNCS, vol. 512, pp. 133–147. Springer, Heidelberg (1991)
10. Kindler, E.: A compositional partial order semantics for Petri net components. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 235–252. Springer, Heidelberg (1997)

11. Leifer, J.J., Milner, R.: Deriving bisimulation congruences for reactive systems. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 243–258. Springer, Heidelberg (2000)
12. Massuthe, P., Reisig, W., Schmidt, K.: An operating guideline approach to the SOA. Annals of Mathematics, Computing & Teleinformatics 1(3), 35–43 (2005)
13. Miculan, M., Yemane, K.: A unifying model of variables and names. In: Sassone, V. (ed.) FOSSACS 2005. LNCS, vol. 3441, pp. 170–186. Springer, Heidelberg (2005)
14. Milner, R.: Communicating and Mobile Systems: the π-Calculus. Cambridge University Press, Cambridge (1999)
15. Milner, R.: Bigraphs for petri nets. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) ACPN 2003. LNCS, vol. 3098, pp. 686–701. Springer, Heidelberg (2004)
16. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, i and ii. Information and Computation 100(1), 1–77 (1992)
17. Milner, R., Sangiorgi, D.: Barbed bisimulation. In: Kuich, W. (ed.) ICALP 1992. LNCS, vol. 623, pp. 685–695. Springer, Heidelberg (1992)
18. Montanari, U., Sassone, V.: Dynamic congruence vs. progressing bisimulation for ccs. Fundamenta Informaticae 16(1), 171–199 (1992)
19. Pistore, M., Sangiorgi, D.: A partition refinement algorithm for the π-calculus. Information and Computation 164(2), 264–321 (2001)
20. Rathke, J., Sassone, V., Sobocinski, P.: Semantic barbs and biorthogonality. In: Seidl, H. (ed.) FOSSACS 2007. LNCS, vol. 4423, pp. 302–316. Springer, Heidelberg (2007)
21. Sangiorgi, D.: A theory of bisimulation for the π-calculus. Acta Informatica 33(1), 69–97 (1996)
22. Sassone, V., Sobociński, P.: A congruence for Petri nets. In: Petri Nets and Graph Transformation. E.N.T.C.S, vol. 127, pp. 107–120. Elsevier, Amsterdam (2005)