

SYMBOLIC AND ASYNCHRONOUS SEMANTICS VIA NORMALIZED COALGEBRAS*

FILIPPO BONCHI^a AND UGO MONTANARI^b

^a ENS Lyon, Université de Lyon, LIP (UMR 5668 CNRS ENS Lyon UCBL INRIA)
e-mail address: filippo.bonchi@ens-lyon.fr

^b Department of Informatics, University of Pisa
e-mail address: ugo@di.unipi.it

ABSTRACT. The operational semantics of interactive systems is usually described by labeled transition systems. Abstract semantics (that is defined in terms of bisimilarity) is characterized by the final morphism in some category of coalgebras. Since the behaviour of interactive systems is for many reasons infinite, symbolic semantics were introduced as a mean to define smaller, possibly finite, transition systems, by employing symbolic actions and avoiding some sources of infiniteness. Unfortunately, symbolic bisimilarity has a different shape with respect to ordinary bisimilarity, and thus the standard coalgebraic characterization does not work. In this paper, we introduce its coalgebraic models.

We will use as motivating examples two asynchronous formalisms: open Petri nets and asynchronous pi-calculus. Indeed, as we have shown in a previous paper, asynchronous bisimilarity can be seen as an instance of symbolic bisimilarity.

INTRODUCTION

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

Universal Coalgebra [40] provides a categorical framework where the behaviour of dynamical systems can be characterized as *final semantics*. More precisely, if $\mathbf{Coalg}_{\mathbf{B}}$ (i.e., the category of \mathbf{B} -coalgebras and \mathbf{B} -cohomomorphisms for a certain endofunctor \mathbf{B}) has a final object, then the behavior of a \mathbf{B} -coalgebra is defined as a final morphism. Intuitively, a final object is a universe of abstract behaviors and a final morphism is a function mapping each system in its abstract behavior. Ordinary LTSS can be represented as coalgebras for a suitable functor. Then, two states are bisimilar if and only if they are identified by a

1998 ACM Subject Classification: F.3.2.

Key words and phrases: Symbolic Semantics, Coalgebras, Process Calculi, Petri nets.

* This paper is an extended version of [9].

^a This work was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme.

^b Research supported in part by IST-FP7-FET open-IP project ASCENS .

final morphism. The image of a certain LTS through a final morphism is its minimal representative (with respect to bisimilarity), which in the finite case can be computed via the partition refinement algorithm [26]. Existence and construction of the minimal transition system is a key property of the coalgebraic approach. It allows to model check efficiently for several properties by eliminating redundant states once and for all. In fact most model checking logics are *adequate*, namely either a formula holds in both the given system and in its minimal representative or it does not hold in both of them.

When bisimilarity is not a congruence, the abstract semantics is defined either as the *largest congruence contained in bisimilarity* [31] or as the *largest bisimulation that is also a congruence* [36]. In this paper we focus on the latter and we call it *saturated bisimilarity* (\sim^S). Indeed it coincides with ordinary bisimilarity on the *saturated transition system* that is obtained from the original LTS by adding the transition $p \xrightarrow{c,a} q$, for every context c , whenever $c(p) \xrightarrow{a} q$.

Many interesting abstract semantics are defined in this way. For example, since late and early bisimilarity of the π -calculus [33] are not preserved under substitution (and thus under input prefixes), in [41] Sangiorgi introduces *open bisimilarity* as the largest bisimulation on π -calculus agents which is closed under substitutions. Other noteworthy examples are asynchronous π -calculus [1, 25], mobile ambients calculus [12, 30] and (explicit [43]) fusion calculus [37]. The definition of saturated bisimilarity as ordinary bisimilarity on the saturated LTS often makes infinite the portion of LTS reachable by any nontrivial agent and, in any case, is very inefficient, since it introduces a large number of additional states and transitions. Inspired by Hennessy and Lin [24], who introduced a *symbolic* semantics of value passing calculi, Sangiorgi defines in [41] a symbolic transition system and symbolic bisimilarity that efficiently characterizes open bisimilarity. After this, many formalisms have been equipped with a symbolic semantics.

In [8], we have introduced a general model that describes at an abstract level both saturated and symbolic semantics. In this abstract setting, a symbolic transition $p \xrightarrow{c,\alpha}_\beta p'$ means that $c(p) \xrightarrow{\alpha} p'$ and c is a smallest context that allows p to perform such a transition. Moreover, a certain *derivation relation* \vdash amongst the transitions of a system is defined: $p \xrightarrow{c_1,\alpha_1} p_1 \vdash p \xrightarrow{c_2,\alpha_2} p_2$ means that the latter transition is a logical consequence of the former. In this way, if all and only the saturated transitions are logical consequences of symbolic transitions, then saturated bisimilarity can be retrieved via the symbolic LTS.

Unfortunately, the ordinary bisimilarity over the symbolic transition system differs from saturated bisimilarity. Symbolic bisimilarity is thus defined with an asymmetric shape: in the bisimulation game, when a player proposes a transition, the opponent can answer with a move with a different label. For example in the open π -calculus, a transition $p \xrightarrow{[a=b],\tau} p'$ can be matched by $q \xrightarrow{\tau} q'$. Moreover, the bisimulation game does not restart from p' and q' , but from p' and $q'\{b/a\}$.

For this reason, ordinary coalgebras fail to characterize symbolic bisimilarity. Here, we provide coalgebraic models for it by relying on the framework of [8].

Consider the example of open bisimilarity discussed above. The fact that open bisimulation does not relate the arriving states p' and q' , but p' and $q'\{b/a\}$, forces us to look for models equipped with an algebraic structure. In [42], *bialgebras* are introduced as a both

algebraic and coalgebraic model, while an alternative approach based on *structured coalgebras*, i.e., on coalgebras in categories of algebras, is presented in [13]. In this paper we adopt the latter and we introduce $\mathbf{Coalg}_{\mathbf{H}}$ (Section 6), a category of structured coalgebras where the saturated transition system can be naively modeled in such a way that \sim^S coincides with the kernel of a final morphism. Then, we focus only on those \mathbf{H} -coalgebras whose sets of transitions are closed w.r.t. the derivation relation \vdash . These form the category of *saturated coalgebras* $\mathbf{Coalg}_{\mathbf{S}_T}$ (Section 7.1) that is (isomorphic to) a covariety of $\mathbf{Coalg}_{\mathbf{H}}$. Thus, it has a final object and bisimilarity coincides with the one in $\mathbf{Coalg}_{\mathbf{H}}$.

In order to characterize symbolic bisimilarity, we introduce the notions of *redundant transition* and *semantically redundant transition*. Intuitively, a transition $p \xrightarrow{c_2, \alpha_2} q$ is redundant if there exists another transition $p \xrightarrow{c_1, \alpha_1} p_1$ that logically implies it, that is $p \xrightarrow{c_1, \alpha_1} p_1 \vdash p \xrightarrow{c_2, \alpha_2} q$; it is *semantically redundant*, if it is “redundant up to bisimilarity”, i.e., $p \xrightarrow{c_1, \alpha_1} p_1 \vdash p \xrightarrow{c_2, \alpha_2} p_2$ and q is bisimilar to p_2 . Now, in order to retrieve saturated bisimilarity by disregarding redundant transitions, we have to remove from the saturated transition system not only all the redundant transitions, but also the semantically redundant ones. This is done in the category of *normalized coalgebras* $\mathbf{Coalg}_{\mathbf{N}_T}$ (Section 7.2). These are defined as coalgebras without redundant transitions. Thus, by definition, a final coalgebra in $\mathbf{Coalg}_{\mathbf{N}_T}$ has no *semantically* redundant transitions.

We prove that $\mathbf{Coalg}_{\mathbf{S}_T}$ and $\mathbf{Coalg}_{\mathbf{N}_T}$ are isomorphic (Section 7.3). This means that a final morphism in the latter category still characterizes \sim^S , but with two important differences w.r.t. $\mathbf{Coalg}_{\mathbf{S}_T}$. First of all, in a final \mathbf{N}_T -coalgebra, there are no semantically redundant transitions. Intuitively, a final \mathbf{N}_T -coalgebra is a universe of *abstract symbolic behaviours* and a final morphism maps each system in its abstract symbolic behaviour. Secondly, minimization in $\mathbf{Coalg}_{\mathbf{N}_T}$ is feasible, while in $\mathbf{Coalg}_{\mathbf{S}_T}$ is not, because saturated coalgebras have all the redundant transitions. Minimizing in $\mathbf{Coalg}_{\mathbf{N}_T}$ coincides with a *symbolic minimization algorithm* that we have introduced in [10] (Section 8). The algorithm shows another peculiarity of normalized coalgebras: minimization relies on the algebraic structure. Since in bialgebras bisimilarity abstracts away from this, we can conclude that our normalized coalgebras are not bialgebras. This is the reason why we work with structured coalgebras.

As motivating examples we will show open Petri nets [27, 3] (Section 2) and asynchronous π -calculus [25, 1] (Section 1). In [8], we have shown that asynchronous bisimilarity [1] is an instance of symbolic bisimilarity. Indeed, in the definition of asynchronous bisimulation, the input transition $p \xrightarrow{a(b)} p'$ can be matched either by $q \xrightarrow{a(b)} q'$ or by $q \xrightarrow{\tau} q'$. In the latter case, the bisimulation game does not restart from p' and q' but from p' and $q'|\bar{a}b$. Thus our framework will provide, as lateral result, also a coalgebraic model for asynchronous bisimilarity that, as far as we know, has never been proposed so far.

In Section 4 and 5 we report the framework of [8] and we recall the basic notions on (structured) coalgebras. In Section 3 we introduce a further example aimed at clarifying the whole framework (by avoiding all the technical details of open Petri nets and asynchronous π). All proofs are in Appendix.

Previous works. Our work relies on the framework introduced in [8] and on the minimization algorithm in [10]. In this work we focus on the coalgebraic characterization of them that appeared in [9]. The present paper extends [9] by (1) introducing the example of asynchronous π -calculus, (2) by adding all the proofs, (3) by explaining in full details the

relationship with the minimization algorithm in [10]. Normalized coalgebras have been previously introduced in [7] for giving a coalgebraic characterization of the theory of reactive systems by Leifer and Milner [29].

1. ASYNCHRONOUS π -CALCULUS

Asynchronous π -calculus has been introduced in [25] for modeling distributed systems interacting via *asynchronous* message passing. Differently from the synchronous case, where messages are sent and received at the same time, in the asynchronous communication, messages are sent and travel through some media until they reach the destination. Therefore sending messages is non blocking (i.e., a process can send messages even if the receiver is not ready to receive), while receiving is blocking (processes must wait until the message has arrived). This asymmetry is reflected on the observations: since sending is non blocking, receiving is unobservable.

In this section, we introduce asynchronous π -calculus and two definitions of bisimilarity (\sim^1 and \sim^a) that, as proved in [1], coincide. In Section 4, we will show that the first is an instance of our general definition of saturated bisimilarity (Definition 4.2) while the second of symbolic bisimilarity (Definition 4.11).

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

$$p ::= \bar{a}b, p_1|p_2, \nu a.p, !g, m \quad m ::= \mathbf{0}, \alpha.p, m_1 + m_2 \quad \alpha ::= a(b), \tau$$

The main difference with the ordinary π -calculus [33] is that here output prefixes are missing. The occurrence of an unguarded $\bar{a}b$ can be thought of as message b that is available on some communication media named a . This message is received whenever it disappears, i.e., it is consumed by some process performing an input. Thus the action of sending happens when $\bar{a}b$ becomes unguarded.

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 $\text{fn}(p)$) 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 $Act = \{a(b), \bar{a}b, \bar{a}(b), \tau \mid a, b \in \mathcal{N}\}$ (ranged over by μ) where b is a *bound name* (written $b \in \text{bn}(\mu)$) in $a(b)$ and $\bar{a}(b)$. In all the other cases a and b are free in μ ($a, b \in \text{fn}(\mu)$). By $\text{nm}(\mu)$ we denote the set of both free and bound names of μ .

The labeled transition system (LTS) 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, i.e., we have implicitly assumed the rule

$$\frac{p \xrightarrow{\mu} q \quad p \equiv_\alpha p'}{p' \xrightarrow{\mu} q}.$$

The main difference with the synchronous case is in the notion of *observation*. Since sending messages is non-blocking, then an external observer can just send messages to a system without knowing if they will be received or not. For this reason the receiving action is not observable and the abstract semantics is defined disregarding input transitions.

$$\begin{array}{lll}
(\text{TAU}) \tau.p \xrightarrow{\tau} p & (\text{IN}) a(b).p \xrightarrow{a(c)} p\{c/b\} & (\text{OUT}) \bar{a}b \xrightarrow{\bar{a}b} \mathbf{0} \\
(\text{COM}) \frac{p \xrightarrow{\bar{a}b} p' \quad q \xrightarrow{a(b)} q'}{p|q \xrightarrow{\tau} p'|q'} & (\text{SUM}) \frac{p \xrightarrow{\mu} p'}{p+q \xrightarrow{\mu} p'} & (\text{PAR}) \frac{p \xrightarrow{\mu} p'}{p|q \xrightarrow{\mu} p'|q} \quad \text{bn}(\mu) \cap \text{fn}(q) = \emptyset \\
(\text{OPN}) \frac{p \xrightarrow{\bar{a}b} p'}{\nu b.p \xrightarrow{\bar{a}(b)} p'} \quad b \neq a & (\text{REP}) \frac{m!m \xrightarrow{\mu} q}{!m \xrightarrow{\mu} q} & (\text{CLS}) \frac{p \xrightarrow{\bar{a}(b)} p' \quad q \xrightarrow{a(b)} q'}{p|q \xrightarrow{\tau} \nu b.p'|q'} \quad b \notin \text{fn}(q) \\
(\text{RES}) \frac{p \xrightarrow{\mu} p'}{\nu b.p \xrightarrow{\mu} \nu b.p'} \quad b \notin \text{nm}(\mu) & &
\end{array}$$

Table 1: Operational semantics of asynchronous π -calculus.

As in the case of the standard π -calculus, in the bisimulation game we have to take care of the bound names in output actions. Indeed, when a process $p \xrightarrow{\bar{a}(b)} p'$, the name b is initially bound in p and becomes free in p' . Thus, in order to avoid name-clashes, in the bisimulation game when comparing p and q , we require b to be *fresh*, namely, different from all the free names of p and q . In the following definitions, by “ $\text{bn}(\mu)$ is fresh” we mean that if μ has a bound name, then it is fresh.

Definition 1.1 ($\sigma\tau$ -Bisimilarity). A symmetric relation R is an $\sigma\tau$ -bisimulation iff, whenever pRq :

- if $p \xrightarrow{\mu} p'$ where μ is not an input action and $\text{bn}(\mu)$ is fresh, then $\exists q'$ such that $q \xrightarrow{\mu} q'$ and $p'Rq'$.

We say that p and q are $\sigma\tau$ -bisimilar (written $p \sim^{\sigma\tau} q$) if and only if there exists an $\sigma\tau$ -bisimulation relating them.

Note that $a(x).\bar{y}x \sim^{\sigma\tau} a(x).\bar{d}x$, even if the two processes are really different when they are put in parallel with a process $\bar{a}b$. In order to obtain an abstract semantics preserved under parallel composition, we proceed analogously to saturated bisimilarity (that we will show in Definition 4.2), i.e., at any step of the bisimulation we put the process in parallel with all possible outputs.

Definition 1.2 (1-Bisimilarity). A symmetric relation R is an 1-bisimulation iff, $\forall \bar{a}b$, whenever pRq ,

- if $\bar{a}b|p \xrightarrow{\mu} p'$ where μ is not an input action and $\text{bn}(\mu)$ is fresh, then $\exists q'$ such that $\bar{a}b|q \xrightarrow{\mu} q'$ and $p'Rq'$.

We say that p and q are 1-bisimilar (written $p \sim^1 q$) if and only if there exists an 1-bisimulation relating them.

The above definition is not very efficient since it considers a quantification over all possible output in parallel. Instead of considering all possible output contexts, we could also consider the input actions. This leads to the following notion of syntactic bisimulation.

Definition 1.3 (Syntactic Bisimilarity). A symmetric relation R is a *syntactic bisimulation* iff, whenever pRq :

- if $p \xrightarrow{\mu} p'$ where $\text{bn}(\mu)$ is fresh, then $\exists q'$ such that $q \xrightarrow{\mu} q'$ and $p'Rq'$.

We say that p and q are syntactic bisimilar (written $p \sim^{SYN} q$) if and only if there exists a syntactic bisimulation relating them.

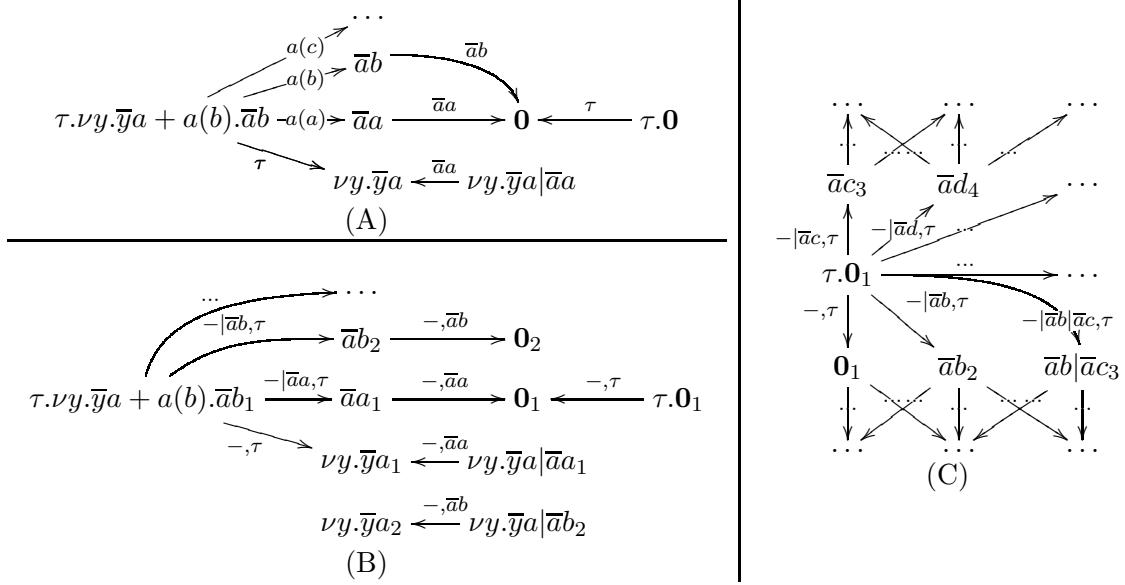


Figure 1: (A) Part of the infinite LTS of $\tau.\nu y.\bar{y}a + a(b).\bar{a}b$ and the LTS of $\tau.\mathbf{0}$. (B) The symbolic transition system α of $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$ and $\tau.\mathbf{0}_1$. (C) Part of the infinite saturated transition system of $\tau.\mathbf{0}_1$.

Note that syntactic bisimilarity is strictly included into 1-bisimilarity. Indeed,

$$\tau + a(b).\bar{a}b \sim^1 \tau, \text{ but } \tau + a(b).\bar{a}b \not\sim^{SYN} \tau.$$

The former equivalence can be understood by observing that both processes can perform a τ transition in any possible context and, when inserted into the context $-\bar{a}x$, both can perform a τ transition going into $\bar{a}x$. More generally, it holds that for all processes $p \sim^1 q \sim^1 r$:

$$\tau.p + a(b).(\bar{a}b|q) \sim^1 \tau.r$$

For instance, by taking $q = r = \mathbf{0}$ and $p = \nu y.\bar{y}a$ (that is 1-bisimilar to $\mathbf{0}$, since both cannot move), we have that $\tau.\nu y.\bar{y}a + a(b).\bar{a}b \sim^1 \tau.\mathbf{0}$. Their LTSS are shown in Figure 1(A).

In order to efficiently characterize \sim^1 , without considering all possible contexts, we have to properly tackle the input transitions.

Definition 1.4 (Asynchronous Bisimilarity). A symmetric relation R is an *asynchronous bisimulation* iff whenever pRq ,

- if $p \xrightarrow{\mu} p'$ where μ is not an input action and $\text{bn}(\mu)$ is fresh, then $\exists q'$ such that $q \xrightarrow{\mu} q'$ and $p'Rq'$,
- if $p \xrightarrow{a(b)} p'$, then $\exists q'$ such that either $q \xrightarrow{a(b)} q'$ and $p'Rq'$, or $q \xrightarrow{\tau} q'$ and $p'R(q'|\bar{a}b)$.

We say that p and q are asynchronous bisimilar (written $p \sim^a q$) if and only if there is an asynchronous bisimulation relating them.

For instance, the symmetric closure of the following relation is an asynchronous bisimulation.

$$R = \{(\tau.\nu y.\bar{y}a + a(b).\bar{a}b, \tau.\mathbf{0}), (\nu y.\bar{y}a, \mathbf{0})\} \cup \{(\bar{a}x, \nu y.\bar{y}a|\bar{a}x) \mid x \in \mathcal{N}\}$$

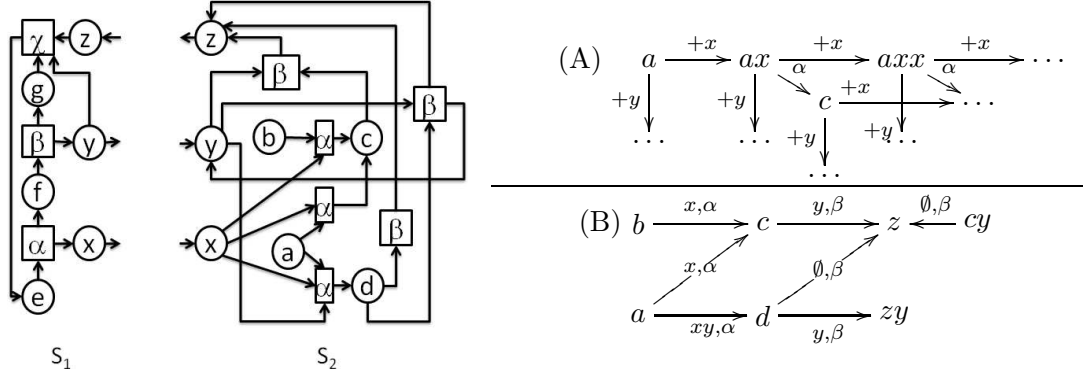


Figure 2: S_1 and S_2 are two open Petri nets. (A) Part of the infinite transition system of $\langle S_2, a \rangle$. (B) The symbolic transition system of $\langle S_2, a \rangle$, $\langle S_2, b \rangle$ and $\langle S_2, cy \rangle$.

In [1], it is proved that $\sim^1 = \sim^a$. In Section 4 we will show that this result is an instance of a more general theorem (Theorem 4.12), since \sim^1 is an instance of saturated bisimilarity and \sim^a is an instance of symbolic bisimilarity. The main contribute of this paper is to give coalgebraic characterization to saturated and symbolic semantics and thus we will characterize both \sim^1 and \sim^a via coalgebras.

2. OPEN PETRI NETS

Differently from process calculi, Petri nets do not have a widely known interactive behavior. Indeed they model concurrent systems that are closed, in the sense that they do not interact with the environment. *Open nets* [27, 3] are P/T Petri nets [39] 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 finite function from X to ω (the set of natural numbers) 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 \emptyset to denote respectively both the empty set and the empty multiset. In order to make lighter the notation we will use aab to denote the multiset $\{a, a, b\}$. Sometimes we will use $a^n b^m$ to denote the multisets containing n copies of a and m copies of b .

Definition 2.1 (Open Net). An *open net* is a tuple $N = (S, T, pre, post, l, I, O)$ where S is the set of places, T is the set of transitions (with $S \cap T = \emptyset$), $pre, post : T \rightarrow S^\oplus$ are functions mapping each transition to its pre- and post-set, $l : T \rightarrow \Lambda$ is a labeling function (Λ is a set of labels) and $I, O \subseteq S$ are the sets of input and output places. A *marked open net* (shortly, *marked net*) is pair $\langle N, m \rangle$ where N is an open net and $m \in S^\oplus$ is a marking.

It is worth noting that standard P/T Petri nets can be thought of as open nets whose sets I and O are empty. Figure 2 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 S_1 , x and y are output places, while z is the only input place. In S_2 , it is the converse. The

$$\begin{array}{l}
\text{(TR)} \frac{t \in T \quad \lambda(t) = l \quad m = \bullet t \oplus c}{N, m \xrightarrow{l} N, t \bullet \oplus c} \quad \text{(IN)} \frac{i \in I_N}{N, m \xrightarrow{\pm i} N, m \oplus i} \quad \text{(OUT)} \frac{o \in O_N \quad o \in m}{N, m \xrightarrow{-o} N, m \ominus o}
\end{array}$$

Table 2: Operational Semantics of marked open nets.

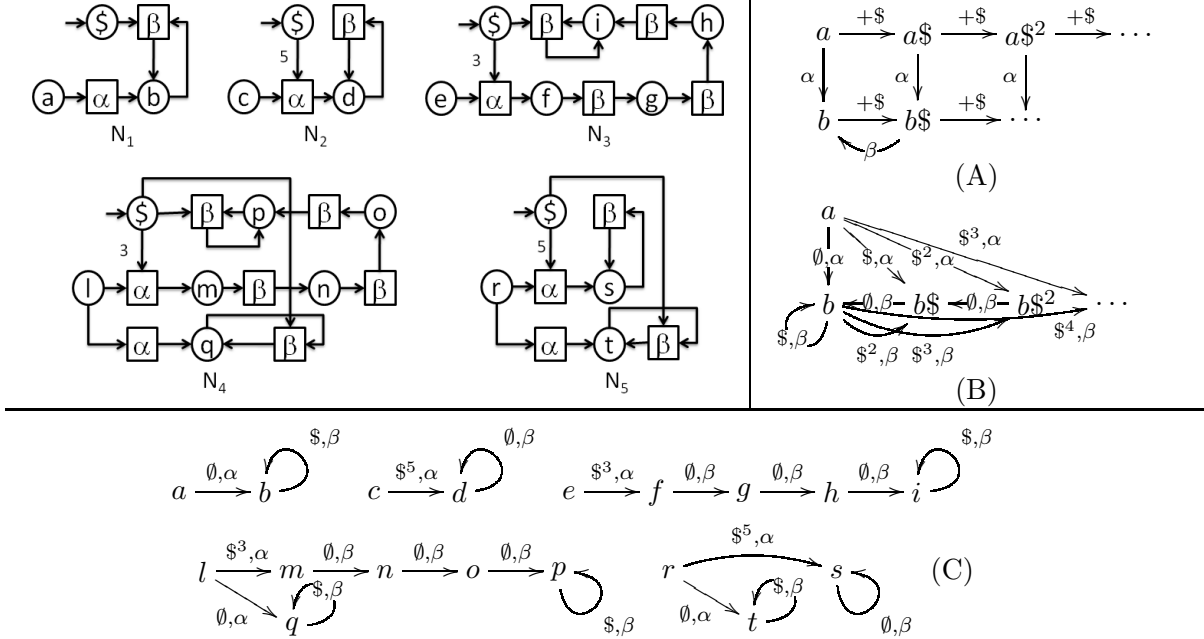


Figure 3: The open nets N_1, N_2, N_3, N_4 and N_5 . (A) Part of the infinite transition system of $\langle N_1, a \rangle$. (B) Part of the infinite saturated transition system of $\langle N_1, a \rangle$. (C) The symbolic transition systems of $\langle N_1, a \rangle, \langle N_2, c \rangle, \langle N_3, e \rangle, \langle N_4, l \rangle$ and $\langle N_5, r \rangle$.

parallel composition of two nets is defined by attaching them on their input and output places. As an example, we can compose S_1 and S_2 by attaching them through x, y and z .

The operational semantics of marked open nets is expressed by the rules on Table 2, where we use $\bullet t$ and $t \bullet$ to denote $pre(t)$ and $post(t)$ and we avoid putting bracket around the marked net $\langle N, m \rangle$, in order to make lighter the notation. 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. Figure 2(A) shows part of the infinite transition system of $\langle S_2, a \rangle$.

The abstract semantics is defined in [2] 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 in inserting the system in all the possible contexts and observing what happens.

In the remainder of the paper we will use as running example the open nets in Figure 3. Since all the places have different names (with the exception of $\$$), in order to make lighter

the notation, we write only the marking to mean the corresponding marked net, e.g. $b^2\$$ means the marked net $\langle N_1, b^2\$ \rangle$.

The marked net a (i.e., $\langle N_1, a \rangle$) represents a system that provides a service β . After the activation α , it provides β whenever the client pay one $\$$ (i.e., the environment insert a token into $\$$). The marked net c instead requires five $\$$ during the activation, but then provides the service β for free. The marked net e , requires three $\$$ during the activation. For three times, the service β is performed for free and then it costs one $\$$. It is easy to see that all these marked nets are not bisimilar. Indeed, a client that has only one $\$$ can have the service β only with a , while a client with five $\$$ can have the service β for six times only with c . The marked net r represents a system that offers the behaviour of both a and c , i.e., either the activation α is for free and then the service β costs one, or the activation costs five and then the service is for free. Also this marked net is different from all the others.

Now consider the marked net l . It offers the behaviour of both a and e , but it is equivalent to a , i.e., $l \sim^N a$. Roughly, the behaviour of e is absorbed by the behaviour of a . This is analogous to what happens in the asynchronous π -calculus where it holds that $a(x).(\bar{a}x \mid p) + \tau.p \sim^1 \tau.p$.

The definition of \sim^N involves an infinite transition system and thus it is often hard to check. As in the case of \sim^1 for the asynchronous π -calculus, we would like to efficiently characterize it. In the following we show an efficient characterization of \sim^N , that we have introduced in [8]. Here and in the rest of the paper, to make simpler the presentation we restrict to open nets with only input places. The general case, is completely analogous and can be found in [8, 4].

First of all, we have to define a *symbolic transition system* that, analogously to the operational semantics of the asynchronous π , performs input-transitions only when needed. We call it η .

Intuitively, the symbolic transition $N, m \xrightarrow{i, \lambda}_\eta N, m'$ is possible if and only if $N, m \oplus i \xrightarrow{\lambda} N, m'$ and i is the smallest multiset (on input places) allowing such transition. This transition system is formally defined by the following rule.

$$\frac{t \in T \quad l(t) = \lambda \quad m = (m \cap \bullet t) \oplus n \quad i \subseteq I^\oplus \quad \bullet t = (m \cap \bullet t) \oplus i}{N, m \xrightarrow{i, \lambda}_\eta N, t^\bullet \oplus n}$$

The marking $m \cap \bullet t$ contains all the tokens of m that are needed to perform the transition t . The marking n 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_2 in Figure 3 with marking $cd\2 and let t be the only transition labeled with α . We have that $cd\$^2 \cap \bullet t = c\2 , $n = d$ and $i = \3 . Thus $N_2, cd\$^2 \xrightarrow{\$^3, \alpha}_\eta N_2, dd$, meaning that $cd\2 needs $\3 to perform α and going into dd . Figure 3(C) shows some symbolic transition systems.

Note that analogously to \sim^{SYN} for the asynchronous π -calculus, the ordinary definition of bisimilarity on the symbolic transition systems for nets, does not coincide with \sim^N . Indeed the symbolic transition systems of a and l in Figure 3(C) are not bisimilar, but as discussed above, $a \sim^N l$. In order to efficiently characterize \sim^N , we have to introduce the following definition.

Definition 2.2 (Net-symbolic Bisimilarity). A symmetric relation R is a *net-symbolic bisimulation* iff, whenever $\langle N_1, m_1 \rangle R \langle N_2, m_2 \rangle$:

- if $\langle N_1, m_1 \rangle \xrightarrow{i, \lambda}_{\eta} \langle N_1, m'_1 \rangle$, then exists a marking m'_2 and $\exists j, k \in I^{\oplus}$ such that:
 - (a) $i = j \oplus k$,
 - (b) $\langle N_2, m_2 \rangle \xrightarrow{j, \lambda}_{\eta} \langle N_2, m'_2 \rangle$ and
 - (c) $\langle N_1, m'_1 \rangle R \langle N_2, m'_2 \oplus k \rangle$.

Two marked nets are net-symbolic bisimilar (written \sim^{NS}) whenever there is a symbolic bisimulation relating them.

For instance, the symmetric closure of the following relation is a net-symbolic bisimulation.

$$R = \{(l, a), (q, b), (m, b\$^3), (n, b\$^2), (o, b\$), (p, b)\}$$

In [8], we have shown that $\sim^N = \sim^{NS}$. In Section 4, we will show that the former is an instance of saturated bisimilarity, while the latter is an instance of symbolic bisimilarity. In Section 7.1 and 7.2, we will give a coalgebraic characterization of both \sim^N and \sim^{NS} by mean of saturated and normalized coalgebras.

3. A SIMPLE WORDS CALCULUS

In the next section we will show a theoretical framework encompassing both asynchronous π -calculus and open Petri nets. In this section, we introduce a *simple words calculus* (swc) as a further instance of the framework presented in the next section. The aim of this “toy calculus” is to provide a more gentle example of the concepts that will be introduced afterward, by avoiding all the technicalities that arise with “real formalisms” like asynchronous π -calculus and open Petri nets.

Let A be an alphabet of symbols (ranged over by a, b, c, \dots) and A^* be the set of finite words over A (ranged over by u, v, w, \dots). We use ε to denote the empty word and uv to denote the concatenation of the words u and v . The set of processes is defined by the following grammar (where $u \in A^*$).

$$p ::= \mathbf{0}, \quad u.p, \quad p_1 + p_2$$

A configuration is a pair $u \triangleright p$ where u is a word (in A^*) representing some *resources* and p a process (generated by the above grammar). The set of all configurations (ranged over by $\gamma_1, \gamma_2, \dots$) is denoted by $Conf$. The algebra \mathbb{W} has as carrier-set $Conf$ and as operators the words $v \in A^*$. The function $v_{\mathbb{W}} : Conf \rightarrow Conf$ maps each configuration $u \triangleright p$ into $uv \triangleright p$. Intuitively, $v_{\mathbb{W}}$ represents a context where configurations can be inserted: the effect of this insertion is that of adding v (via word-concatenation) to the resources of the configuration. This is analogous to asynchronous π -calculus and open nets. There, resources are respectively outputs (in parallel) and tokens (in input places). Moreover, in those formalisms the environment can arbitrarily add new resources (via context composition).

Differently from asynchronous π and open nets, in swc all the transitions are labeled with the same observation \bullet . Therefore, we fix the set of *observations* of swc to be $O_{\mathcal{W}} = \{\bullet\}$ (the subscript \mathcal{W} will be useful later to distinguish the observations of swc from those of asynchronous π and open nets). The operational semantics of swc is given by the *transition*

relation $tr_{\mathcal{W}} \subseteq Conf \times O_{\mathcal{W}} \times Conf$ defined by the following rules (together with the symmetric one for $+$).

$$uv \triangleright u.p \xrightarrow{\bullet} uv \triangleright p \quad \frac{u \triangleright p \xrightarrow{\bullet} u \triangleright p'}{u \triangleright p + q \xrightarrow{\bullet} u \triangleright p'}$$

Intuitively, the process $u.p$ needs the resources u in order to evolve. If u is present in the configuration (as a suffix) then, $u.p$ becomes p . Note that, differently from asynchronous π and open nets, the resources u are not consumed, but only “read” (we have chosen to give this read-behavior to swc , just for simplifying the following examples).

Definition 3.1 (Saturated Bisimilarity for swc). Let $R \subseteq Conf \times Conf$ be a symmetric relation. R is a *saturated bisimulation* iff, $\forall v \in A^*$, whenever $\gamma_1 R \gamma_2$:

- $v_{\mathbb{W}}(\gamma_1) R v_{\mathbb{W}}(\gamma_2)$,
- if $\gamma_1 \xrightarrow{\bullet} \gamma'_1$, then $\exists \gamma'_2$ such that $\gamma_2 \xrightarrow{\bullet} \gamma'_2$ and $\gamma'_1 R \gamma'_2$.

We write $\gamma_1 \sim^S \gamma_2$ iff there is a saturated bisimulation R such that $\gamma_1 R \gamma_2$.

For instance, the configurations $ab \triangleright ab.\mathbf{0}$ and $ab \triangleright \varepsilon.\mathbf{0}$ are saturated bisimilar, because for any word v both $abv \triangleright ab.\mathbf{0}$ and $abv \triangleright \varepsilon.\mathbf{0}$ can only perform one transition and then stop. A more interesting example is the following. For all words $u, v \in A^*$ such that $v = uw$ (i.e., u is a prefix of v), it holds that

$$\varepsilon \triangleright u.p + v.p \sim^S \varepsilon \triangleright u.p$$

because for any word $v' \in A^*$, $v' \triangleright u.p + v.p$ and $v' \triangleright u.p$ have the same behaviour. For those v' having u as prefix (i.e., $v' = uw'$), both the configurations can only perform transitions going into $v' \triangleright p$; for those v' where u is not a prefix, both the configurations stop. As it happens for the asynchronous π -calculus and open nets, the behaviour of $v.p$ is somehow “absorbed” by the behaviour of $u.p$. By joining the two previous examples, we have that:

$$\varepsilon \triangleright a.ab.\mathbf{0} + ab.\varepsilon.\mathbf{0} \sim^S \varepsilon \triangleright a.ab.\mathbf{0}$$

Indeed, for all the words $v' \in A^*$ having ab as a prefix (i.e., $v' = abw'$) the configuration $abw' \triangleright a.ab.\mathbf{0} + ab.\varepsilon.\mathbf{0}$ can go either in $abw' \triangleright ab.\mathbf{0}$ or in $abw' \triangleright \varepsilon.\mathbf{0}$, while the configuration $abw' \triangleright a.ab.\mathbf{0}$ can only go in $abw' \triangleright ab.\mathbf{0}$ that, as shown in our first example, is bisimilar to $abw' \triangleright \varepsilon.\mathbf{0}$. For all the other words, the two configuration behave exactly in the same way.

For simplifying the explanation, it is useful to introduce the *saturated transition system*: $u \triangleright p \xrightarrow{v, \bullet}_S u' \triangleright p'$ iff $uv \triangleright p \xrightarrow{\bullet} u' \triangleright p'$. It is easy to see that the standard notion of bisimilarity on this transition system coincides with \sim^S . The saturated transition systems of $\varepsilon \triangleright u.p + v.p$ and $\varepsilon \triangleright u.p$ are shown in Figure 4(A). For making lighter the notation, in that figure and in the following ones we have omitted the observation \bullet . Note that $\varepsilon \triangleright u.p + v.p$ and $\varepsilon \triangleright u.p$ perform the same saturated transitions (and thus they are saturated bisimilar, as discussed above).

In order to give a more efficient characterization of \sim^S (that avoids the quantification over all words $v \in A^*$), we define a *symbolic transition system* that, like the saturated transition system, is labeled with pairs v, \bullet (for $v \in A^*$). The main difference is that a symbolic transition $u \triangleright p \xrightarrow{v, \bullet}_\omega u' \triangleright p'$ is performed only when v is the “minimal word” such that $uv \triangleright p \xrightarrow{\bullet} u' \triangleright p'$. The symbolic transition system $\omega \subseteq Conf \times A^* \times O_{\mathcal{W}} \times Conf$ is

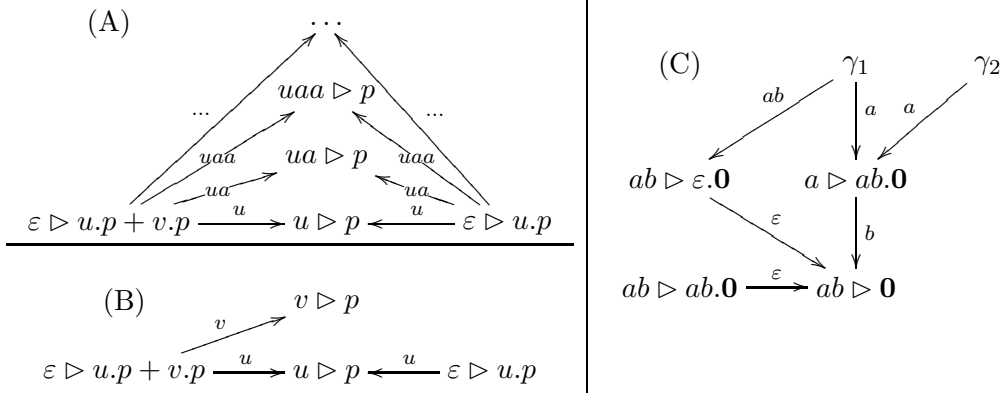


Figure 4: (A) The saturated transition systems of $\varepsilon \triangleright u.p + v.p$ and $\varepsilon \triangleright u.p$. (B) The symbolic transition systems of $\varepsilon \triangleright u.p + v.p$ and $\varepsilon \triangleright u.p$. (C) The symbolic transition systems of $\gamma_1 = \varepsilon \triangleright a.ab.0 + ab.\varepsilon.0$ and $\gamma_2 = \varepsilon \triangleright a.ab.0$.

defined by the following rules (together with the symmetric rule for $+$).

$$uv \triangleright u.p \xrightarrow{\varepsilon, \bullet}_{\omega} uv \triangleright p \quad u \triangleright uv.p \xrightarrow{v, \bullet}_{\omega} uv \triangleright p \quad \frac{u \triangleright p \xrightarrow{v, \bullet}_{\omega} u' \triangleright p'}{u \triangleright p + q \xrightarrow{v, \bullet}_{\omega} u' \triangleright p'}$$

In the central rule, the process $uv.p$ needs the resources uv to evolve. In the configuration, there are only u resources and thus the process “takes from the environment” the word v . In the leftmost rule, all the needed resources (u) are already present in the configuration (as a prefix) and thus the process can evolve without taking resources from the environment (i.e., by taking ε). The symbolic transition systems of $\varepsilon \triangleright u.p + v.p$ and $\varepsilon \triangleright u.p$ are depicted in Figure 4(B). Note that the former process can perform one symbolic transition more than the latter, even if they perform the same saturated transitions. The symbolic transition systems of $\gamma_1 = \varepsilon \triangleright a.ab.0 + ab.\varepsilon.0$ and $\gamma_2 = \varepsilon \triangleright a.ab.0$ are shown in Figure 4(C).

Note that the standard notion of bisimilarity defined over $\xrightarrow{v, \bullet}_{\omega}$ (hereafter called *syntactic bisimilarity* and denoted by \sim^W) is strictly included into \sim^S . For example, $\varepsilon \triangleright u.p$ and $\varepsilon \triangleright u.p + v.p$ (with u prefix of v) are in \sim^S but not in \sim^W because $\varepsilon \triangleright u.p + v.p \xrightarrow{v, \bullet}_{\omega} v \triangleright p$, while $\varepsilon \triangleright u.p$ only performs a symbolic transition labeled with u . The same holds for $\varepsilon \triangleright a.ab.0 + ab.\varepsilon.0$ and $\varepsilon \triangleright a.ab.0$.

In order to capture \sim^S by exploiting the symbolic transition system we need a more elaborated notion of bisimulation that relies on an *inference system*. For better explaining it, observe that the following “monotonicity property” holds:

$$\forall v \in A^* \text{ and } \forall u \triangleright p, u' \triangleright p' \in Conf, \text{ if } u \triangleright p \xrightarrow{\bullet} u' \triangleright p', \text{ then } uv \triangleright p \xrightarrow{\bullet} u'v \triangleright p'.$$

This property states that when adding the resources v to the original configuration (or, equivalently, when inserting the configuration into the context $v_{\mathbb{W}}(-)$), all the transitions of the original configuration are preserved. This is analogous to what happens in the asynchronous π -calculus (where putting outputs in parallel does not inhibit any transition) and in open Petri nets (where inserting tokens in input places does not inhibit any transition).

An inference system is a set of rules stating properties like those just described. For the case of swc, the inference system $T_{\mathbb{W}}$ is defined by the following rule (parametric w.r.t.

$v \in A^*$).

$$\frac{\gamma \xrightarrow{\bullet} \gamma'}{v_{\mathbb{W}}(\gamma) \xrightarrow{\bullet} v_{\mathbb{W}}(\gamma')}$$

This rule just states the above monotonicity property. Moreover, it induces a *derivation relation* $\vdash_{T_{\mathbb{W}}} \subseteq (Conf \times A^* \times O_{\mathbb{W}} \times Conf) \times (Conf \times A^* \times O_{\mathbb{W}} \times Conf)$ as follows:

$$\gamma \xrightarrow{v, \bullet} \gamma' \vdash_{T_{\mathbb{W}}} \gamma \xrightarrow{vw, \bullet} w_{\mathbb{W}}(\gamma')$$

Consider the saturated transitions of $\varepsilon \triangleright u.p + v.p$ in Figure 4(A) and fix $\gamma = \varepsilon \triangleright u.p + v.p$. We have that $(\gamma \xrightarrow{u, \bullet}_S u \triangleright p) \vdash_{T_{\mathbb{W}}} (\gamma \xrightarrow{ua, \bullet}_S ua \triangleright p) \vdash_{T_{\mathbb{W}}} (\gamma \xrightarrow{uaa, \bullet}_S uaa \triangleright p) \vdash_{T_{\mathbb{W}}} \dots$. More generally, $\forall w \in A^*$,

$$\gamma \xrightarrow{u, \bullet}_S u \triangleright p \vdash_{T_{\mathbb{W}}} \gamma \xrightarrow{uw, \bullet}_S uw \triangleright p$$

and in the case of $\gamma = \varepsilon \triangleright u.p + v.p$ in Figure 4(B), this means that

$$\gamma \xrightarrow{u, \bullet}_{\omega} u \triangleright p \vdash_{T_{\mathbb{W}}} \gamma \xrightarrow{v, \bullet}_{\omega} v \triangleright p.$$

This is somehow useful to understand the causes of the mismatch between \sim^S and \sim^W (syntactic bisimilarity). First, observe that symbolic transitions can derive through $T_{\mathbb{W}}$ all and only the saturated transitions (this will be formally shown in the next section). Then, recall that the configurations $\varepsilon \triangleright u.p + v.p$ and $\varepsilon \triangleright u.p$ are in \sim^S because can perform the same saturated transitions, but they are not in \sim^W because the former can perform the symbolic transition $\xrightarrow{v, \bullet}_{\omega}$. This symbolic transition is *redundant* since it can be derived from $\xrightarrow{u, \bullet}_{\omega}$ through the inference system $T_{\mathbb{W}}$. More explicitly, all the saturated transitions that can be derived from $\xrightarrow{v, \bullet}_{\omega}$ can also be derived from $\xrightarrow{u, \bullet}_{\omega}$ and thus $\xrightarrow{v, \bullet}_{\omega}$ does not add any meaningful information about the saturated behaviour of the configuration. We can avoid this problem by employing the following notion of bisimulation.

Definition 3.2 (Symbolic Bisimilarity for swc). Let $R \subseteq Conf \times Conf$ be a symmetric relation. R is a *symbolic bisimulation* iff whenever $\gamma_1 R \gamma_2$:

- if $\gamma_1 \xrightarrow{v, \bullet}_{\omega} \gamma'_1$, then $\exists \gamma'_2, \gamma''_2 \in Conf, u \in A^*$ s.t. $\gamma_2 \xrightarrow{u, \bullet}_{\omega} \gamma'_2$, $\gamma_2 \xrightarrow{u, \bullet}_{\omega} \gamma''_2 \vdash_{T_{\mathbb{W}}} \gamma_2 \xrightarrow{v, \bullet}_{\omega} \gamma''_2$ and $\gamma'_1 R \gamma''_2$.

We write $\gamma_1 \sim^{SYM} \gamma_2$ iff there is a symbolic bisimulation R such that $\gamma_1 R \gamma_2$.

For example $\varepsilon \triangleright u.p + v.p \sim^{SYM} \varepsilon \triangleright u.p$ (when $v = uw$), because if $\varepsilon \triangleright u.p + v.p \xrightarrow{v, \bullet}_{\omega} u \triangleright p$, then $\varepsilon \triangleright u.p \xrightarrow{u, \bullet}_{\omega} u \triangleright p$ and this transition derives $\varepsilon \triangleright u.p \xrightarrow{uw, \bullet}_{\omega} w_{\mathbb{W}}(u \triangleright p)$ that is $\varepsilon \triangleright u.p \xrightarrow{v, \bullet}_{\omega} v \triangleright p$.

For an example of symbolic bisimulation, take $\gamma_1 = \varepsilon \triangleright a.ab.\mathbf{0} + ab.\varepsilon.\mathbf{0}$ and $\gamma_2 = \varepsilon \triangleright a.ab.\mathbf{0}$ in Figure 4(C) and consider the symmetric closure of the following relation.

$$R = \{(\gamma_1, \gamma_2), (a \triangleright ab.\mathbf{0}, a \triangleright ab.\mathbf{0}), (ab \triangleright \varepsilon.\mathbf{0}, ab \triangleright ab.\mathbf{0}), (ab \triangleright \mathbf{0}, ab \triangleright \mathbf{0})\}$$

For the last three pairs, it is easy to check that the configurations satisfy the above requirements. For (γ_1, γ_2) , this is more interesting: the transition $\gamma_1 \xrightarrow{ab, \bullet}_{\omega} ab \triangleright \varepsilon.\mathbf{0}$ can be matched by $\gamma_2 \xrightarrow{a, \bullet}_{\omega} a \triangleright ab.\mathbf{0}$ because, by definition of $\vdash_{T_{\mathbb{W}}}$, $\gamma_2 \xrightarrow{a, \bullet}_{\omega} a \triangleright ab.\mathbf{0} \vdash_{T_{\mathbb{W}}} \gamma_2 \xrightarrow{ab, \bullet}_{\omega} ab \triangleright ab.\mathbf{0}$ and $(ab \triangleright \varepsilon.\mathbf{0}, ab \triangleright ab.\mathbf{0}) \in R$.

In the next section we will show that $\sim^S = \sim^{SYM}$. Before concluding this section, it is worth to make a final remark. The reader would have thought that in order to retrieve

\sim^S from the symbolic transition system, one could just remove all the “redundant transitions”, i.e., all those symbolic transitions $\gamma \xrightarrow{v, \bullet}_{\omega} \gamma''$ such that there exists another symbolic transition $\gamma \xrightarrow{u, \bullet}_{\omega} \gamma'$ deriving it (in Section 7 this removal will be called *normalization*). It is important to show that this is not enough to retrieve \sim^S : consider the symbolic transition systems of $\gamma_1 = \varepsilon \triangleright a.ab.\mathbf{0} + ab.\varepsilon.\mathbf{0}$ and $\gamma_2 = \varepsilon \triangleright a.ab.\mathbf{0}$ shown in Figure 4(C). They have no redundant transitions, but still $\gamma_1 \sim^S \gamma_2$ and $\gamma_1 \not\sim^W \gamma_2$. The transition $\gamma_1 \xrightarrow{ab, \bullet}_{\omega} ab \triangleright \varepsilon.\mathbf{0}$ is not redundant, because $\gamma_1 \xrightarrow{a, \bullet}_{\omega} a \triangleright ab.\mathbf{0} \not\vdash_{T_{\mathbb{W}}} \gamma_1 \xrightarrow{ab, \bullet}_{\omega} ab \triangleright \varepsilon.\mathbf{0}$, since $b_{\mathbb{W}}(a \triangleright ab.\mathbf{0}) = ab \triangleright ab.\mathbf{0} \neq ab \triangleright \varepsilon.\mathbf{0}$. However, it is *semantically redundant*, because $\gamma_1 \xrightarrow{a, \bullet}_{\omega} a \triangleright ab.\mathbf{0} \vdash_{T_{\mathbb{W}}} \gamma_1 \xrightarrow{ab, \bullet}_{\omega} ab \triangleright ab.\mathbf{0}$ and the states $ab \triangleright ab.\mathbf{0}$ and $ab \triangleright \varepsilon.\mathbf{0}$ are semantically equivalent (i.e., $ab \triangleright ab.\mathbf{0} \sim^S ab \triangleright \varepsilon.\mathbf{0}$).

In order to characterize \sim^S through ω , we should eliminate all the semantically redundant transitions, but this is impossible without knowing a priori \sim^S . This is the main motivation for the introduction of *normalized coalgebras* in Section 7.

4. SATURATED AND SYMBOLIC SEMANTICS

In Section 1 and Section 2, we have introduced asynchronous π -calculus and open Petri nets. In both cases, their abstract semantics is defined in two different ways: either by inserting the systems into all possible contexts (like \sim^1 and \sim^N) or by inserting the system only in those contexts that are really needed (like \sim^a and \sim^{NS}). Moreover, the latter coincides with the former and thus can be thought as an efficient characterization of the former.

This sort of “double definition” of the abstract semantics recurs in many formalisms modeling interactive systems, such as mobile ambients [12], open π -calculus [41] and explicit fusion calculus [43]. In [8], we have introduced a theoretical framework that generalizes this “double definition” and encompasses all the above mentioned formalisms. In this section we recall this framework by employing as running examples the simple words calculus, the asynchronous π -calculus and open Petri nets.

4.1. Saturated Semantics. Given a small category \mathbf{C} , a $\Gamma(\mathbf{C})$ -algebra is an algebra for the algebraic specification in Figure 5 where $|\mathbf{C}|$ denotes the set of objects of \mathbf{C} , $\|\mathbf{C}\|$ the set of arrows of \mathbf{C} and, for all $i, j \in |\mathbf{C}|$, $\mathbf{C}[i, j]$ denotes the set of arrows from i to j . Thus, a $\Gamma(\mathbf{C})$ -algebra \mathbb{X} consists of a $|\mathbf{C}|$ -sorted family $X = \{X_i \mid i \in |\mathbf{C}|\}$ of sets and a function $c_{\mathbb{X}} : X_i \rightarrow X_j$ for all $c \in \mathbf{C}[i, j]$. Moreover, these functions must satisfy the equations in Figure 5: $id_{i_{\mathbb{X}}}$ is the identity function on X_i and if $d; e = c$ in \mathbf{C} , $(d; e)_{\mathbb{X}}$ is equal to $c_{\mathbb{X}}$.¹ Hereafter, we will use $\int X$ to denote the set of the *elements* of a $\Gamma(\mathbf{C})$ -algebra \mathbb{X} , namely, the disjoint union $\sum_{i \in |\mathbf{C}|} X_i$.

The main definition of the framework presented in [8] is that of *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 4.1 (Context Interactive System). A *context interactive system* \mathcal{I} is a quadruple $\langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ where:

- \mathbf{C} is a small category,

¹Note that $\Gamma(\mathbf{C})$ -algebras coincide with functors from \mathbf{C} to \mathbf{Set} and $\Gamma(\mathbf{C})$ -homomorphisms coincide with natural transformations amongst functors. Thus, $\mathbf{Alg}_{\Gamma(\mathbf{C})}$ is isomorphic to $\mathbf{Set}^{\mathbf{C}}$ (the category of covariant presheaves over \mathbf{C}).

$$\begin{aligned}
 \text{specification } \Gamma(\mathbf{C}) = & \\
 \text{sorts} & \\
 i & \quad \forall i \in |\mathbf{C}| \\
 \text{operations} & \\
 c : i \rightarrow j & \quad \forall c \in \mathbf{C}[i, j] \\
 \text{equations} & \\
 id_i(x) = x & \\
 e(d(x)) = c(x) & \quad \forall d; e = c
 \end{aligned}$$

 Figure 5: Algebraic specification $\Gamma(\mathbf{C})$.

- \mathbb{X} is a $\Gamma(\mathbf{C})$ -algebra,
- O is a set of observations,
- $tr \subseteq \int X \times O \times \int X$ is a labeled transition relation ($p \xrightarrow{o} p'$ means $(p, o, p') \in tr$).

Intuitively, objects of \mathbf{C} are interfaces of the system, while arrows are contexts. Every element p of X_i represents a state with interface i and it can be inserted into the context $c \in \mathbf{C}[i, j]$, obtaining a new state $c_{\mathbb{X}}(p)$ that has interface j . 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 [8] we proposed a general notion of bisimilarity that generalizes the abstract semantics of a large variety of formalisms [12, 1, 41, 37, 44, 11]. 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 transitions.

Definition 4.2 (Saturated Bisimilarity). Let $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ be a context interactive system. Let $R = \{R_i \subseteq X_i \times X_i \mid i \in |\mathbf{C}|\}$ be a $|\mathbf{C}|$ -sorted family of symmetric relations. R is a *saturated bisimulation* iff, $\forall i, j \in |\mathbf{C}|, \forall c \in \mathbf{C}[i, j]$, whenever pR_iq :

- $c_{\mathbb{X}}(p) R_j c_{\mathbb{X}}(q)$,
- if $p \xrightarrow{o} p'$ with $p' \in X_k$ for some $k \in |\mathbf{C}|$, then $\exists q' \in X_k$ such that $q \xrightarrow{o} q'$ and $p' R_k q'$.

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

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 $c_{\mathbb{X}}(p) \xrightarrow{o} q$. Trivially the ordinary bisimilarity over SATTS coincides with \sim^S .

Proposition 4.3. \sim^S is the coarsest bisimulation congruence.

A Context Interactive Systems for swc. In Section 3, we have introduced a simple words calculus. Here we show its context interactive system $\mathcal{W} = \langle \mathbf{Wor}, \mathbb{W}, O_{\mathcal{W}}, tr_{\mathcal{W}} \rangle$. Recall that ε is the empty word and that uv denote the concatenation of the words u and v . The category \mathbf{Wor} is defined as follows:

- $|\mathbf{Wor}| = \{\circ\}$;
- $\mathbf{Wor}[\circ, \circ] = A^*$;
- $id_{\circ} = \varepsilon$;
- $\forall u, v \in A^*, u; v = uv$.

The algebra \mathbb{W} , the set of observations $O_{\mathcal{W}}$ and the transition relation $tr_{\mathcal{W}}$ have been already introduced in Section 3. In swc , all the configurations have the same interface (sort) and thus, in the category **Wor** there is only one object. It is easy to see that saturated bisimilarity for swc (Definition 3.1) is an instance of Definition 4.2.

A Context Interactive Systems for open Petri nets. In the following we formally define $\mathcal{N} = \langle \mathbf{Tok}, \mathbb{N}, \Lambda, tr_{\mathcal{N}} \rangle$ that is the context interactive system of all open nets (labeled over the set of labels Λ). Let Pl be an infinite set. We assume that the input places of all open nets are taken from Pl . Formally, we assume that if I is the set of input places of an open net N , then $I \in \mathbf{P}(Pl)$ (where $\mathbf{P}(Pl)$ denotes the powerset of Pl).

The category **Tok** is formally defined as follows:

- $|\mathbf{Tok}| = \{I \mid I \in \mathbf{P}(Pl)\}$;
- $\forall I, J \in |\mathbf{Tok}|$, if $I = J$ then $\mathbf{Tok}[I, J] = I^{\oplus}$ while, if $I \neq J$ then $\mathbf{Tok}[I, J] = \emptyset$;
- $\forall I \in |\mathbf{Tok}|$, $id_I = \emptyset$;
- $\forall i_1, i_2 \in I^{\oplus}$, $i_1; i_2 = i_1 \oplus i_2$.

Intuitively objects are sets of places I . Arrows $i : I \rightarrow I$ are multisets of tokens on I , while there exists no arrow $i : I \rightarrow J$ for $I \neq J$. Composition of arrows is just the sum of multisets and, obviously, the identity arrow is the empty multiset.

We say that a marked open net $\langle N, m \rangle$ has interface I if the set of input places of N is I . For example the marked open net $\langle N_1, a \rangle$ has interface $\{\$ \}$. Let us define the $\Gamma(\mathbf{Tok})$ -algebra \mathbb{N} . For any sort I , the carrier set N_I contains all the marked open nets with interface I . For any operator $i \in \mathbf{Tok}[I, I]$, the function $i_{\mathbb{N}}$ maps $\langle N, m \rangle$ into $\langle N, m \oplus i \rangle$.

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 2. The saturated transition system of $\langle N_1, a \rangle$ is shown in Figure 3(B).

Proposition 4.4. *Let $\langle N_1, m_1 \rangle$ and $\langle N_2, m_2 \rangle$ be two marked nets both with interface I . Thus $\langle N_1, m_1 \rangle \sim^N \langle N_2, m_2 \rangle$ iff $\langle N_1, m_1 \rangle \sim_I^S \langle N_2, m_2 \rangle$.*

A Context Interactive System for asynchronous π . We now introduce the context interactive system $\mathcal{A} = \langle \mathbf{Out}, \mathbb{A}, O_{\mathcal{A}}, tr_{\mathcal{A}} \rangle$ for the asynchronous π -calculus. First, we assume the set of names \mathcal{N} to be in one to one correspondence with ω_0 (the set of natural numbers ω without the number 0). In \mathcal{A} , we use numbers in ω_0 in place of names in \mathcal{N} , but for the sake of readability, in all the concrete examples of processes we use names $a, b, c, \dots \in \mathcal{N}$ thought of as the natural numbers $1, 2, 3, \dots \in \omega_0$. We need such correspondence, because we use the well order $1 < 2 < 3 \dots$. Given an $n \in \omega$, it denotes both the number and the set of numbers in ω_0 smaller or equal than n . For instance, 2 denotes both the number 2 and the set $\{1, 2\}$ that correspond, respectively, to the name b and to the set of names $\{a, b\}$; while 0 denotes both the number 0 and the empty set: the former does not correspond to any name and the latter corresponds to the empty set of names \emptyset . In the following, we will use the name in \mathcal{N} and numbers in ω_0 interchangeably. Also, when fixed some sets $n, m \dots$ we will use i, j to range over the elements of these sets.

The category of interfaces and contexts is **Out**, formally defined as follows:

- $|\mathbf{Out}| = \omega$;
- if $m \geq n$, then $\mathbf{Out}[n, m]$ is the set of contexts generated by $c ::= -, c[\bar{i}j]$, with $i, j \in m$;
- if $m < n$, then $\mathbf{Out}[n, m] = \emptyset$;
- $\forall n \in \omega$, id_n is $- \in \mathbf{Out}[n, n]$;

- arrows composition is the syntactic composition of contexts.

Note that a context could correspond to several arrows with different sources and targets. For instance, the context $-\overline{1}2$ (corresponding to $-\overline{a}b$) is, e.g., both an arrow $0 \rightarrow 2$ and an arrow $1 \rightarrow 6$. The composition of the arrow $-\overline{1}2 : 0 \rightarrow 2$ with $-\overline{3}4 : 2 \rightarrow 5$ is $-\overline{1}2\overline{3}4 : 0 \rightarrow 5$.

Let us define the $\Gamma(\mathbf{Out})$ -algebra \mathbb{A} . For every object n , A_n is the set of asynchronous π -processes p such that $n \geq \max \text{fn}(p)$. Intuitively in asynchronous π , interfaces are sets of names. A process with interface n uses only names in n (not all, just some). Given a process p and a natural number $n \geq \max \text{fn}(p)$, we denote with p_n the process p with interface n . For instance, there exists several processes corresponding to $\tau.\mathbf{0}$: $\tau.\mathbf{0}_0, \tau.\mathbf{0}_1, \dots$. Each of these is considered different from the others because has a different interface. This may seem a bit strange, but is quite standard in categorical semantics of process calculi [17, 18, 21] as well as in their graphical encodings [32, 19, 5, 20].

Extensively, 0 is the empty interface and A_0 is the set of all π -processes without free names. The set A_1 contains all the processes with free names in $\{1\}$ (corresponding to $\{a\}$) and A_2 contains all the processes with free names in $\{1, 2\}$ (corresponding to $\{a, b\}$) and so on \dots

In order to fully define \mathbb{A} , we still have to specify its operations $c_{\mathbb{A}}$ for all $c \in \mathbf{Out}[n, m]$. Given a process $p \in A_n$, $c_{\mathbb{A}}(p)$ is the process with interface m obtained by syntactically inserting p into c . For instance, $a(x).\overline{x}a_1$ can be inserted into $-\overline{b}c : 1 \rightarrow 3$ obtaining the process $a(x).\overline{x}a|\overline{b}c_3$.

Note that, differently from what happens in open nets, an asynchronous π -process can dynamically enlarge its interface by receiving names in input or *extruding* some restricted name. Name extrusion is an essential feature of the π -calculus that can be easily explained by looking at the rule (OPN) in Table 1: the name b is local (i.e., bound) in $\nu b.p$, but it becomes global (i.e., free) whenever p send it to the environment. In \mathcal{A} , we are going to assume that processes p_n with interface n always extrude the name $n+1$: this ensures that the extruded name is fresh (i.e., $n+1 \notin \text{fn}(p_n)$).

The set of observations is $O_{\mathcal{A}} = \{\overline{ij}, \overline{i}(), \tau|i, j \in \omega_0\}$. 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 transition structure $tr_{\mathcal{A}}$ (denoted by $\rightarrow_{\mathcal{A}}$) is defined by the following rules, where $i, j \in \omega_0$ represent in the premises the corresponding names in \mathcal{N} , while in the conclusion the numbers in ω_0 . Moreover the transition relation in the premise is the one in Table 1.

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

Note that for τ and not-bound output, $\text{fn}(p') \subseteq \text{fn}(p) \subseteq n$, and thus $p' \in A_n$. For the case of bound output instead, the extruded name $n+1$ could occur free in p' . Thus $\text{fn}(p') \subseteq n+1$ and $p' \in A_{n+1}$.

In our context interactive system \mathcal{A} , processes only perform τ and output transitions. The contexts are all the possible outputs. Therefore is almost trivial to see that saturated bisimilarity coincides with \sim^1 . Figure 1(C) shows the saturated transition system of $\tau.\mathbf{0}_1$.

Proposition 4.5. *Let p, q be asynchronous π -processes, and let $n \geq \max \text{fn}(p \cup q)$. Then $p \sim^1 q$ iff $p_n \sim_n^S q_n$.*

4.2. Symbolic Semantics. Saturated bisimulation is a good notion of equivalence but it is hard to check, since it involves a quantification over all contexts. In [8], we have introduced a general notion of *symbolic bisimilarity* that coincides with saturated bisimilarity, but it avoids to consider all contexts. The idea is to define a symbolic transition system where transitions are labeled both with the usual observation and also with the minimal context that allows the transition. First we need to introduce *context transition systems*.

Definition 4.6 (Context Transition System). Given a category \mathbf{C} , a $\Gamma(\mathbf{C})$ -algebra \mathbb{X} and a set of observations O , a *context transition system* $\beta \subseteq \int X \times \|\mathbf{C}\| \times O \times \int X$ is a transition relation labeled with $\|\mathbf{C}\| \times O$ ($p \xrightarrow{c,o}_\beta p'$ means that $(p, c, o, p') \in \beta$).

An example of context transition system is η defined in Section 2: each transition is labeled with both a multiset of tokens i and an observation λ . Also the saturated transition system is a context transition systems. Hereafter, given a context transition system β , we will write $\xrightarrow{c,o}_\beta$ to denote the transitions of β , $\xrightarrow{c,o}_S$ to denote the saturated transitions and $\xrightarrow{c,o}$ (without subscript) to denote the transitions of the *total context transition system* $t = \int X \times \|\mathbf{C}\| \times O \times \int X$.

Definition 4.7 (Inference System). Given a category \mathbf{C} , a $\Gamma(\mathbf{C})$ -algebra \mathbb{X} and a set of observations O , an inference system T is a set of rules of the following format, where $i, j \in |\mathbf{C}|$, $o, o' \in O$, $c \in \mathbf{C}[i, i']$ and $d \in \mathbf{C}[j, j']$.

$$\frac{p_i \xrightarrow{o} q_j}{c(p_i) \xrightarrow{o'} d(q_j)}$$

In this rule, i, j, o, o', c and d are constants, while p_i and q_j are variables ranging over X_i and X_j , respectively. Therefore, the above rule states that all processes with interface i that perform a transition with observation o going into a state q_j with interface j , when inserted into the context c can perform a transition with the observation o' going into $d(q_j)$. In other words, this rule is in a (multisorted) SOS format, where the operators (here, contexts) are unary and there is only one transition in the premise of the rules. Note that, however, this kind of rules is not intended to be used for expressing the operational semantics of a formalism (as in the case of SOS), but instead for describing “useful properties” about how contexts modify the behaviour of systems.

In the following, we write $c \xrightarrow{o}_{o'} d$ to mean a rule like the above. The rules $c \xrightarrow{o}_{o'} c'$ and $d \xrightarrow{o'}_{o''} d'$ derive the rule $c; d \xrightarrow{o}_{o''} c'; d'$ if $c; d$ and $c'; d'$ are defined. Given an inference system T , $\Phi(T)$ is the set of all the rules derivable from T together with the identities rules ($\forall o \in O$ and $\forall i, j \in |\mathbf{C}|$, $id_i \xrightarrow{o}_{o} id_j$).

Definition 4.8 (Derivations). Let \mathbf{C} be a category, \mathbb{X} be a $\Gamma(\mathbf{C})$ -algebra, O be a set of observations. An inference system T defines a *derivation relation* $\vdash_T \subseteq t \times t$ amongst the transitions of the total context transition system.

We say that $p \xrightarrow{c_1, o_1} p_1$ *derives* $p \xrightarrow{c_2, o_2} p_2$ (written $p \xrightarrow{c_1, o_1} p_1 \vdash_T p \xrightarrow{c_2, o_2} p_2$) if there exist $d, e \in \|\mathbf{C}\|$ such that $d \xrightarrow{o_1} e \in \Phi(T)$, $c_1; d = c_2$ and $e_{\mathbb{X}}(p_1) = p_2$.

Note that the above definition can be extended to the transitions of any pairs of context transition systems β_1, β_2 : $p \xrightarrow{c_1, o_1}_{\beta_1} p_1 \vdash_T p \xrightarrow{c_2, o_2}_{\beta_2} p_2$ iff $p \xrightarrow{c_1, o_1} p_1 \vdash_T p \xrightarrow{c_2, o_2} p_2$.

Until now, context transition systems and inference systems are not related with the transitions relations tr of context interactive systems. The following definition makes a link between them.

Definition 4.9 (Soundness and Completeness). Let $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ be a context interactive system, β a context transition system and T an inference system.

We say that β and T are *sound* w.r.t. \mathcal{I} iff

$$\text{if } p \xrightarrow{c', o'}_{\beta} q' \text{ and } p \xrightarrow{c', o'}_{\beta} q' \vdash_T p \xrightarrow{c, o} q, \text{ then } p \xrightarrow{c, o}_S q.$$

We say that β and T are *complete* w.r.t. \mathcal{I} iff

$$\text{if } p \xrightarrow{c, o}_S q, \text{ then there exists } p \xrightarrow{c', o'}_{\beta} q' \text{ such that } p \xrightarrow{c', o'}_{\beta} q' \vdash_T p \xrightarrow{c, o} q.$$

Definition 4.10. Let $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ be a context interactive system, β a context transition system and T an inference system. If β and T are sound and complete w.r.t. \mathcal{I} we say that β is a *symbolic transition system* (SCTS for short) for \mathcal{I} .

For instance, the saturated transition system η (defined in Section 2 for open nets) is a symbolic transition system (this will be formally stated in Proposition 4.16). Also the saturated transition system is a symbolic transition system (take as T the empty inference system), while the total context transition system is usually not sound.

A symbolic transition system could be considerably smaller than the saturated transition system, but still containing all the information needed to recover \sim^S . Note that the ordinary bisimilarity over SCTS (hereafter called *syntactic bisimilarity* and denoted by \sim^W) is usually strictly included in \sim^S . As an example consider the marked open nets a and l . These are not syntactically bisimilar, since $l \xrightarrow{\S^3, \alpha}_{\eta} m$ while a cannot (Figure 3(C)). However, they are saturated bisimilar, since $\sim^S = \sim^N$. Analogously, the ordinary bisimilarity over the LTS of the asynchronous π does not coincide with \sim^1 : $a(b).\bar{a}b + \tau$ and τ are 1-bisimilar, but not syntactically bisimilar (at the end of this section, we will show that also the transition system of asynchronous π in Table 1 is somehow a SCTS).

In literature, several SCTS are defined in [41, 37, 44]. In these works, transitions are labeled with both “fusions” of names and the ordinary labels. Other noteworthy examples are the IPOs and the borrowed contexts of [29] and [16]: here all the transitions are labeled only with the minimal contexts and the observations can be thought as τ s. Also in all these cases, syntactic bisimilarity is too fine grained. In order to recover \sim^S through the symbolic transition system we need a more elaborated definition of bisimulation.

Definition 4.11 (Symbolic Bisimilarity). Let $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ be an interactive system, T be a set of rules and β be a context transition system. Let $R = \{R_i \subseteq X_i \times X_i \mid i \in |\mathbf{C}|\}$ be a $|\mathbf{C}|$ -sorted family of symmetric relations. R is a *symbolic bisimulation* iff $\forall i \in |\mathbf{C}|$, whenever pR_iq :

- if $p \xrightarrow{c, o}_{\beta} p'$, then $\exists c_1, o_1, q'_1, q'$ such that $q \xrightarrow{c_1, o_1}_{\beta} q'_1$ and $q \xrightarrow{c_1, o_1}_{\beta} q'_1 \vdash_T q \xrightarrow{c, o} q'$ and $p'R_kq'$.

We write $p \sim_i^{SYM} q$ iff there exists a symbolic bisimulation R such that pR_iq .

Theorem 4.12. *Let \mathcal{I} be a context interactive system, β a context transition system and T an inference system. If β and T are sound and complete w.r.t. \mathcal{I} , then $\sim^{SYM} = \sim^S$.*

Symbolic Semantics for swc. The symbolic transition system ω and the inference system $T_{\mathcal{W}}$ for swc have already been defined in Section 3. It is also easy to see that symbolic bisimilarity for swc (Definition 3.2) is an instance of Definition 4.11. Therefore, in order to apply Theorem 4.12, we only need to prove that ω and $T_{\mathcal{W}}$ are sound and complete.

Proposition 4.13. *ω and $T_{\mathcal{W}}$ are sound and complete w.r.t. \mathcal{W} .*

Corollary 4.14 (From Theorem 4.12). *In swc, $\sim^S = \sim^{SYM}$.*

Symbolic Semantics for open Petri nets. The symbolic transition system for open Petri nets is η defined in Section 2. The inference system $T_{\mathcal{N}}$ is defined by the following rule parametric w.r.t. $\lambda \in \Lambda$, $I \in \mathbf{P}(Pl)$ and $i \in I^{\oplus}$.

$$\frac{N, m \xrightarrow{\lambda} N, m'}{N, m \oplus i \xrightarrow{\lambda} N, m' \oplus i}$$

Its intuitive meaning is that for all possible observations λ and multiset i on input places, if a marked net performs a transition with observation λ , then the addition of i preserves this transition.

Now, consider derivations between transitions of open nets. It is easy to see that $N, m \xrightarrow{i_1, \lambda_1} N, m_1 \vdash_{T_{\mathcal{N}}} N, m \xrightarrow{i_2, \lambda_2} N, m_2$ if and only if $\lambda_2 = \lambda_1$ and there exists a multiset x on the input places of N such that $i_2 = i_1 \oplus x$ and $m_2 = m_1 \oplus x$. For all the nets N_k of Figure 3, this just means that for all observations λ and for all multisets m, n , we have that $\langle N_k, m \rangle \xrightarrow{\S^i, \lambda}_{\eta} \langle N_k, n \rangle \vdash_{T_{\mathcal{N}}} \langle N_k, m \rangle \xrightarrow{\S^{i+j}, \lambda} \langle N_k, n \S^j \rangle$. From this observation, it is easy to see that the definition of net-symbolic bisimilarity is an instance of symbolic bisimilarity.

Proposition 4.15. *Let $\langle N_1, m_1 \rangle$ and $\langle N_2, m_2 \rangle$ be two marked nets both with interface I . Thus $\langle N_1, m_1 \rangle \sim^{NS} \langle N_2, m_2 \rangle$ iff $\langle N_1, m_1 \rangle \sim_I^{SYM} \langle N_2, m_2 \rangle$.*

Thus, in order to prove that $\sim^N = \sim^{NS}$, we have only to prove that η and $T_{\mathcal{N}}$ are sound and complete w.r.t. \mathcal{N} and then apply the general Theorem 4.12.

Proposition 4.16. *η and $T_{\mathcal{N}}$ are sound and complete w.r.t. \mathcal{N} .*

Corollary 4.17 (From Theorem 4.12). *$\sim^N = \sim^{NS}$.*

Symbolic Semantics for asynchronous π . In the case of asynchronous π -calculus, the ordinary LTS closely corresponds to the SCTS that we are going to introduce. The transitions labeled with an input $a(b)$ are substantially transitions saying that if the process is inserted into $-|\bar{a}b$, then it can perform a τ . The symbolic transition system α for the asynchronous π -calculus is defined by the following rules, where in the premises there are standard transitions (from Table 1), $i, j \in \omega_0$ represent in the premises the corresponding names in \mathcal{N} , while in the conclusion the numbers in ω_0 and $- \in \mathbf{Out}[n, n]$ and $-|\bar{i}m \in \mathbf{Out}[n, n']$.

$$\frac{p \xrightarrow{\tau} p'}{p_n \xrightarrow{-i, \tau}_{\alpha} p'_n} \quad \frac{p \xrightarrow{\bar{i}j} p'}{p_n \xrightarrow{-i, \bar{i}j}_{\alpha} p'_n} \quad \frac{p \xrightarrow{\bar{i}(n+1)} p'}{p_n \xrightarrow{-i, \bar{i}()}_{\alpha} p'_{n+1}} \quad \frac{p \xrightarrow{i(m)} p' \quad n' = \max\{m, n\}}{p_n \xrightarrow{-|\bar{i}m, \tau}_{\alpha} p'_{n'}}$$

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 process in parallel. Note that the interface of the arriving state depends on the received name m : if it is smaller than n , then the arriving interface is still n , otherwise it is extended to m (i.e., $\max \text{fn}(p') \leq \max \{m, n\} = n'$).

Part of the SCTS of $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$ and $\tau.\mathbf{0}_1$ are shown in Figure 1(B). There and in the following we avoid to specify the source and the target of the contexts labelling the transitions, since these can be inferred by the sorts of starting and arriving states. As well as the ordinary LTS, the symbolic transition system is infinite, because the input can receive any possible name in \mathcal{N} . It is well known that, instead of considering all possible input names, it is enough to consider only the free names and one fresh name (all the other fresh are useless). By slightly modifying the general definition of the context interactive system \mathcal{A} , we could have defined a symbolic context transition system that only receive in input those names that are strictly needed. We have made a different choice for the following reasons: (a) the presentation of this modified context interactive system is a bit more contrived; (b) the actual presentation is mainly aimed at showing how an input transition “can be matched” by a τ transition (instead of focusing on finite representation); (c) there exists several other sources of infiniteness (discussed in Section 9) that cannot be trivially tackled by our framework.

Let us define an inference system $T_{\mathcal{A}}$ that describes how contexts transform transitions. Since our contexts are just parallel outputs, all the contexts preserve transitions. This is expressed by the following rules parametric w.r.t. $n, m \in \omega$, $i, j \in n$, $c \in \mathbf{Out}[n, m]$.

$$\begin{array}{ccc}
 (\text{TAU}_c) \frac{p_n \xrightarrow{\tau} q_n}{c(p_n) \xrightarrow{\tau} c(q_n)} & (\text{OUT}_c) \frac{p_n \xrightarrow{\bar{i}j} q_n}{c(p_n) \xrightarrow{\bar{i}j} c(q_n)} & (\text{BOUT}_c) \frac{p_n \xrightarrow{\bar{i}()} q_{n+1}}{c(p_n) \xrightarrow{\bar{i}()} c^{+1}(q_{n+1})}
 \end{array}$$

Here, $c^{+1} \in \mathbf{Out}[n+1, m+1]$ is the same syntactic context as c , but with different interfaces.

Derivations amongst transitions of asynchronous π -processes are quite analogous to those amongst open Petri nets. Particularly relevant is the following kind of derivation: for all processes p_n, q_n , for all names $i \in n$ and $j \in m$,

$$p_n \xrightarrow{-i\tau}_\alpha q_n \vdash_{T_{\mathcal{A}}} p_n \xrightarrow{-|\bar{i}j, \tau} q|\bar{i}j_m.$$

Intuitively, this means that in the original LTS, the τ transitions derive the input transitions. Instantiating the general definition of symbolic bisimulation to α and $T_{\mathcal{A}}$, we retrieve the definition of asynchronous bisimulation. Indeed transitions of the form $p \xrightarrow{-i\mu}_\alpha p'$ (in the original LTS, these correspond to τ and output), can be matched only by transitions with the same label, since the context $-$ is not decomposable.

The transitions $p \xrightarrow{-|\bar{i}j, \tau}_\alpha p'$ (corresponding to the input in the original LTS) can be matched either by $q \xrightarrow{-|\bar{i}j, \tau}_\alpha q'$, or by $q \xrightarrow{-i\tau}_\alpha q'$. In other words, when $p \xrightarrow{-|\bar{i}j, \tau}_\alpha p'$, then q can answer with $q \xrightarrow{-i\tau}_\alpha q'$, since $q \xrightarrow{-i\tau}_\alpha q' \vdash_{T_{\mathcal{A}}} q \xrightarrow{-|\bar{i}j, \tau} q'|\bar{i}j$.

Proposition 4.18. *Let p, q be asynchronous π -processes, and let $n \geq \max \text{fn}(p \cup q)$. Then $p \sim^a q$ iff $p_n \sim_n^{SYM} q_n$.*

Therefore \sim^1 is the saturated bisimulation for \mathcal{A} , while \sim^a is its the symbolic version. We can employ our general Theorem 4.12 to prove that $\sim^1 = \sim^a$ by showing that the SCTS α and the inference system $T_{\mathcal{A}}$ are sound and complete w.r.t. \mathcal{A} .

Proposition 4.19. α and $T_{\mathcal{A}}$ are sound and complete w.r.t. \mathcal{A} .

Corollary 4.20 (By Theorem. 4.12). $\sim^1 = \sim^a$ as shown in [1].

5. (STRUCTURED) COALGEBRAS

In this section we recall the basic notions of the theory of coalgebras and the coalgebraic characterization of labeled transition systems and bisimilarity.

Definition 5.1 (Coalgebra). Let $\mathbf{B} : \mathbf{C} \rightarrow \mathbf{C}$ be an endofunctor on a category \mathbf{C} . A **B-coalgebra** is a pair $\langle X, \alpha \rangle$ where X is an object of \mathbf{C} and $\alpha : X \rightarrow \mathbf{B}(X)$ is an arrow. A **B-morphism** $f : \langle X, \alpha \rangle \rightarrow \langle Y, \beta \rangle$ is an arrow $f : X \rightarrow Y$ of \mathbf{C} such that the following diagram commutes. **B-coalgebras** and **B-morphisms** form the category $\mathbf{Coalg}_{\mathbf{B}}$.

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ \alpha \downarrow & & \downarrow \beta \\ \mathbf{B}(X) & \xrightarrow{\mathbf{B}(f)} & \mathbf{B}(Y) \end{array}$$

For instance, labeled transition systems with labels in L are coalgebras for the functor $\mathbf{P}(L \times Id) : \mathbf{Set} \rightarrow \mathbf{Set}$, where \mathbf{Set} denotes the category of sets and functions. This functor maps each set X into the set $\mathbf{P}(L \times X)$ (i.e., the powerset of $L \times X$) and each function $f : X \rightarrow Y$ into $\mathbf{P}(L \times f) : \mathbf{P}(L \times X) \rightarrow \mathbf{P}(L \times Y)$ that, for all $A \in \mathbf{P}(L \times X)$, is defined as $\mathbf{P}(L \times f)(A) = \{(l, f(x)) \text{ s.t. } (l, x) \in A\}$. Concretely, a LTS is a set of states X together with a transition function $\alpha : X \rightarrow \mathbf{P}(L \times X)$ mapping each state into a set of pairs (l, x) representing transitions with labels $l \in L$ and next state $x \in X$. A $\mathbf{P}(L \times Id)$ -morphism is a “zig-zag” morphism, i.e., a function between the sets of states that both preserves and reflects the transitions.

We can think of symbolic transition systems as ordinary $\mathbf{P}(L \times Id)$ -coalgebras where the labels in L are pairs (c, o) (for c a contexts, and o an observation), but this representation is somehow inadequate. Figure 6 shows a function between the states space of two $\mathbf{P}(L \times Id)$ -coalgebras. This is not a $\mathbf{P}(L \times Id)$ -morphism since the transition $\gamma_1 \xrightarrow{ab, \bullet} ab \triangleright \varepsilon.0$ is not preserved. The same holds for the morphisms in Figure 7: these are not $\mathbf{P}(L \times Id)$ -morphisms since the transitions $l \xrightarrow{\S^3, \alpha} m$ and $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1 \xrightarrow{-|\bar{a}a, \tau} \bar{a}a_1$ are not preserved. In Section 7, we will show the category of normalized coalgebras where these maps are morphisms.

Under certain conditions, $\mathbf{Coalg}_{\mathbf{B}}$ has a *final coalgebra* (unique up to isomorphism) into which every **B-coalgebra** can be mapped via a unique **B-morphism**. The final coalgebra can be viewed as the universe of all possible **B-behaviours**: the unique morphism into the final coalgebra maps every state of a coalgebra to a canonical representative of its behaviour. This provides a general notion of behavioural equivalence (hereafter referred to as bisimilarity): two **B-coalgebras** are **B-equivalent** iff they are mapped to the same element of the final coalgebra. Moreover, the image of a coalgebra through the final morphism is its minimal realization w.r.t. bisimilarity. In the finite case, this can be done via a minimization algorithm, that for LTSS coincides with [26].

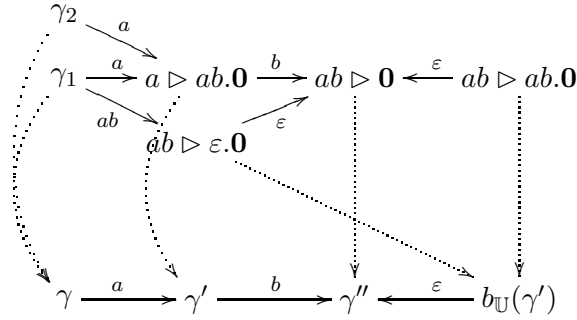


Figure 6: The dotted arrows represent a map between the states space of two transition systems. It is not a $\mathbf{P}(L \times Id)$ -morphism but it is a morphism in the category of normalized coalgebra.

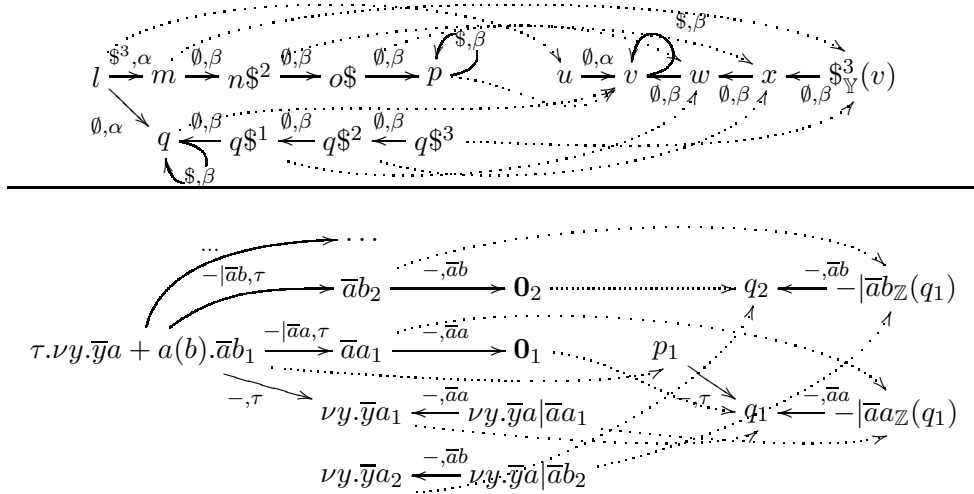


Figure 7: The dotted arrows represent maps between the states space of transition systems. Both are not $\mathbf{P}(L \times Id)$ -morphisms, but they are morphisms in the category of normalized coalgebras.

Unfortunately, due to cardinality reasons, $\mathbf{P}(L \times Id)$ does not have a final object [40]. One satisfactory solution consists in replacing the powerset functor \mathbf{P} by the *countable* powerset functor \mathbf{P}_c , which maps a set to the family of its countable subsets. Then, $\mathbf{P}_c(L \times Id)$ -coalgebras are one-to-one with transition systems with *countable degree*. Unlike the functor $\mathbf{P}(L \times Id)$, the functor $\mathbf{P}_c(L \times Id)$ admits final coalgebras (Example 6.8 of [40]).

The coalgebraic representation using functor $\mathbf{P}_c(L \times Id)$ is not completely satisfactory, because the intrinsic algebraic structure of the states is lost. This calls for the introduction of *structured coalgebras* [14], i.e., coalgebras for an endofunctor on a category \mathbf{Alg}_Γ of algebras for a specification Γ . Since morphisms in a category of structured coalgebras are also Γ -homomorphisms, *bisimilarity* (i.e. the kernel of a final morphism) *is a congruence* w.r.t. the operations in Γ .

Moreover, since we would like that the structured coalgebraic model is compatible with the unstructured, set-based one, we are interested in functors $\mathbf{B}^\Gamma : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Alg}_\Gamma$ that are the *lifting* of some functor $\mathbf{B} : \mathbf{Set} \rightarrow \mathbf{Set}$ along the *forgetful* functor $\mathbf{U}^\Gamma : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Set}$ (i.e., the following diagram commutes).

$$\begin{array}{ccc} \mathbf{Alg}_\Gamma & \xrightarrow{\mathbf{B}^\Gamma} & \mathbf{Alg}_\Gamma \\ \mathbf{U}^\Gamma \downarrow & & \downarrow \mathbf{U}^\Gamma \\ \mathbf{Set} & \xrightarrow{\mathbf{B}} & \mathbf{Set} \end{array}$$

Proposition 5.2 (From [14]). *Let Γ be an algebraic specification. Let $\mathbf{U}^\Gamma : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Set}$ be the forgetful functor. If $\mathbf{B}^\Gamma : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Alg}_\Gamma$ is a lifting of $\mathbf{P}_c(L \times Id)$ along \mathbf{U}^Γ , then (1) $\mathbf{Coalg}_{\mathbf{B}^\Gamma}$ has a final object, (2) bisimilarity is uniquely induced by $\mathbf{P}_c(L \times Id)$ -bisimilarity and (3) bisimilarity is a congruence.*

In [42], *bialgebras* are used as structures combining algebras and coalgebras. Bialgebras are richer than structured coalgebras, in the sense that they can be seen both as coalgebras on algebras and also as algebras on coalgebras. In [14], it is shown that whenever \mathbf{B}^Γ is a lifting of some \mathbf{B} , then \mathbf{B}^Γ -coalgebras are also bialgebras. In Section 7.2, we will introduce *normalized coalgebras* that are structured coalgebras, but not bialgebras (i.e., their endofunctor is not the lifting of some endofunctor on \mathbf{Set}). This is our motivation for using structured coalgebras.

6. COALGEBRAIC SATURATED SEMANTICS

Recall the definition of context interactive system (Definition 4.1). Here, and in the rest of the paper we will always assume to work with a context interactive system $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ where (a) $\|\mathbf{C}\|$ (the set of morphisms of the small category \mathbf{C}) is a countable set and (b) the transition relation tr has countable degree, i.e., the set of transitions outgoing from a state is countable. These two assumptions also guarantee that the saturated transition system has countable degree.

In this section we introduce the coalgebraic model for the saturated transition system. First we model it as a coalgebra over $\mathbf{Set}^{|\mathbf{C}|}$, i.e., the category of $|\mathbf{C}|$ -sorted families of sets and functions. Therefore in this model, all the algebraic structure is missing. Then we lift it to $\mathbf{Alg}_{\Gamma(\mathbf{C})}$ that is the category of $\Gamma(\mathbf{C})$ -algebras and $\Gamma(\mathbf{C})$ -homomorphisms. Recall that when X is a $|\mathbf{C}|$ -sorted family of sets, $\int X = \sum_{i \in |\mathbf{C}|} X_i$.

Definition 6.1. $\mathbf{G} : \mathbf{Set}^{|\mathbf{C}|} \rightarrow \mathbf{Set}^{|\mathbf{C}|}$ is defined for each $|\mathbf{C}|$ -sorted family of set X and for each $i \in |\mathbf{C}|$ as $\mathbf{G}(X_i) = \mathbf{P}_c(\sum_{j \in |\mathbf{C}|} (\mathbf{C}[i, j] \times O \times \int X))$. Analogously for arrows.

A \mathbf{G} -coalgebra is a \mathbf{C} -sorted family $\alpha = \{\alpha_i : X_i \rightarrow \mathbf{G}(X_i) \mid i \in |\mathbf{C}|\}$ of functions assigning to each $p \in X_i$ a set of transitions (c, o, q) where c is an arrow of \mathbf{C} (context) with source i , o is an observation and q is the arriving state. Note that q can have any possible sort ($q \in \int X$).

For each $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$, we define the \mathbf{G} -coalgebra $\langle X, \alpha_{\mathcal{I}} \rangle$ corresponding to the SATTS, where $\forall i \in |\mathbf{C}|, \forall p \in X_i, (c, o, q) \in \alpha_{\mathcal{I}}(p)$ iff $(c_{\mathbb{X}}(p), o, q) \in tr$.

Now we want to define an endofunctor \mathbf{H} on $\mathbf{Alg}_{\Gamma(\mathbf{C})}$ that is a lifting of \mathbf{G} and such that $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ is a \mathbf{H} -coalgebra. In order to do that, we must define how \mathbf{H} modifies the

operations of $\Gamma(\mathbf{C})$ -algebras. This is described by the following rule.

$$\frac{p \xrightarrow{c_1, o} q \quad c_1 = d; c_2}{d(p) \xrightarrow{c_2, o} q}$$

Intuitively, this rule states how to compute the saturated transitions of $d(p)$ from the saturated transitions of p . Indeed, if $p \xrightarrow{d; c_2, o}_S q$, then $d; c_2(p) \xrightarrow{o} q$ and then $d(p) \xrightarrow{c_2, o}_S q$.

Hereafter, in order to make lighter the notation, we will avoid to specify sorts. We will denote a $\Gamma(\mathbf{C})$ -algebra \mathbb{X} as $\langle X, d_{\mathbb{X}}^0, d_{\mathbb{X}}^1, \dots \rangle$ where X is the $|\mathbf{C}|$ -sorted carrier set of \mathbb{X} and $d_{\mathbb{X}}^i$ is the function corresponding to the operator $d^i \in \|\mathbf{C}\|$.

Definition 6.2. $\mathbf{H} : \mathbf{Alg}_{\Gamma(\mathbf{C})} \rightarrow \mathbf{Alg}_{\Gamma(\mathbf{C})}$ maps each $\mathbb{X} = \langle X, d_{\mathbb{X}}^0, d_{\mathbb{X}}^1, \dots \rangle \in \mathbf{Alg}_{\Gamma(\mathbf{C})}$ into $\langle \mathbf{G}(X), d_{\mathbf{H}(\mathbb{X})}^0, d_{\mathbf{H}(\mathbb{X})}^1, \dots \rangle$ where $\forall d \in \|\mathbf{C}\|$, $\forall A \in \mathbf{G}(X)$, $d_{\mathbf{H}(\mathbb{X})}A = \{(c_2, l, x) \mid (c_1, l, x) \in A \text{ and } c_1 = d; c_2\}$. For arrows, it is defined as \mathbf{G} .

Intuitively, $\mathbf{H} : \mathbf{Alg}_{\Gamma(\mathbf{C})} \rightarrow \mathbf{Alg}_{\Gamma(\mathbf{C})}$ can be thought of as an extension of the functor $\mathbf{G} : \mathbf{Set}^{|\mathbf{C}|} \rightarrow \mathbf{Set}^{|\mathbf{C}|}$ to the category $\mathbf{Alg}_{\Gamma(\mathbf{C})}$. Each algebra \mathbb{X} with ($|\mathbf{C}|$ -sorted) carrier set X is mapped to an algebra having as ($|\mathbf{C}|$ -sorted) carrier set $\mathbf{G}(X)$. The elements of $\mathbf{G}(X)$ with sort i are sets of triples (c_1, o, x) (representing sets of transitions) where $c_1 : i \rightarrow j$ is an arrow in \mathbf{C} . For each arrow $d : i \rightarrow k$, there is an operator in $\mathbf{H}(\mathbb{X})$ $d_{\mathbf{H}(\mathbb{X})} : \mathbf{G}(X_i) \rightarrow \mathbf{G}(X_k)$ that maps each set A of triples in $\mathbf{G}(X_i)$ into the set of triples $\{(c_2, l, x) \mid (c_1, l, x) \in A \text{ and } c_1 = d; c_2\}$ (note that the arrows c_2 have source k).

It is worth to note that by definition, \mathbf{H} is a lifting of \mathbf{G} . Thus, by Proposition 5.2, follows that $\mathbf{Coalg}_{\mathbf{H}}$ has final object and that bisimilarity is a congruence.²

In [42], it is shown that every process algebra whose operational semantics is given by GSOS rules, defines a bialgebra. In that approach the carrier of the bialgebra is an initial algebra T_{Σ} for a given algebraic signature Σ , and the GSOS rules specify how an endofunctor \mathbf{B}_{Σ} behaves with respect to the operations of the signature. Since there exists only one arrow $?_{\Sigma} : T_{\Sigma} \rightarrow \mathbf{B}_{\Sigma}(T_{\Sigma})$, to give SOS rules is enough for defining the bialgebra (i.e., $\langle T_{\Sigma}, ?_{\Sigma} \rangle$) and then for assuring compositionality of bisimilarity. Our construction slightly differs from this. Indeed, the carrier of our coalgebra is \mathbb{X} , that is not the initial algebra of $\mathbf{Alg}_{\Sigma(\mathbf{C})}$. Then there might exist several or none structured coalgebras with carrier \mathbb{X} . In the following we prove that $\alpha_{\mathcal{I}} : \mathbb{X} \rightarrow \mathbf{H}(\mathbb{X})$ is a $\Gamma(\mathbf{C})$ -homomorphism.

Theorem 6.3. $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ is a \mathbf{H} -coalgebra.

Now, since a final coalgebra $F_{\mathbf{H}}$ exists in $\mathbf{Coalg}_{\mathbf{H}}$ and since $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ is a \mathbf{H} -coalgebra, there exists a final morphism from $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$. The kernel of this coincides with \sim^S , because (a) \mathbf{H} -bisimilarity coincides with \mathbf{G} -bisimilarity (by Proposition 5.2(2)) and (b) bisimilarity of \mathbf{G} -coalgebras for the saturated transition system coincides with saturated bisimilarity.

By [13], $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ is also a bialgebra (since \mathbf{H} is a lifting). In the next section we will introduce coalgebraic models for symbolic semantics that are structured coalgebras but not bialgebras.

²Proposition 5.2 holds also for many-sorted algebras and many sorted-sets [15].

7. COALGEBRAIC SYMBOLIC SEMANTICS

In Section 6 we have characterized saturated bisimilarity as the equivalence induced by the final morphism from $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ (i.e., the \mathbf{H} -coalgebra corresponding to SATTs) to $F_{\mathbf{H}}$. This is theoretically interesting, but pragmatically useless. Indeed SATTs is usually infinitely branching (or in any case very inefficient), and so is the minimal model. In this section we use symbolic bisimilarity in order to give an efficient and coalgebraic characterization of \sim^S . We provide a notion of *redundant transitions* and we introduce *normalized coalgebras* as coalgebras without redundant transitions. The category of normalized coalgebras ($\mathbf{Coalg}_{\mathbf{N}_T}$) is isomorphic to the category of *saturated coalgebras* ($\mathbf{Coalg}_{\mathbf{S}_T}$) that is (isomorphic to) a full subcategory of $\mathbf{Coalg}_{\mathbf{H}}$ that contains only those coalgebras “satisfying” an inference system T . From the isomorphism follows that \sim^S coincides with the kernel of the final morphism in $\mathbf{Coalg}_{\mathbf{N}_T}$. This provides a characterization of \sim^S really useful: every equivalence class has a canonical model that is smaller than that in $\mathbf{Coalg}_{\mathbf{H}}$ because normalized coalgebras have no redundant transitions. Moreover, minimizing in $\mathbf{Coalg}_{\mathbf{N}_T}$ is usually feasible since it abstracts away from redundant transitions.

7.1. Saturated Coalgebras. Hereafter we refer to a context interactive system $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ and to an inference system T . First, we extend \vdash_T (Definition 4.8) with the operators of $\Gamma(\mathbf{C})$ -algebras.

Definition 7.1 (Extended Derivation). Let \mathbb{X} be a $\Gamma(\mathbf{C})$ -algebra. A transition $p \xrightarrow{c_1, o_1} q_1$ derives a transition $d_{\mathbb{X}}(p) \xrightarrow{c_2, o_2} q_2$ in \mathbb{X} through T (written $(c_1, o_1, q_1) \vdash_{T, \mathbb{X}}^d (c_2, o_2, q_2)$) iff there exist $e, e' \in \|\mathbf{C}\|$ such that $c_1; e = d; c_2$ and $e \xrightarrow{o_1}_{o_2} e' \in \Phi(T)$ and $e'_{\mathbb{X}}(q_1) = q_2$.

Intuitively, $\vdash_{T, \mathbb{X}}^d$ allows to derive from the set of transitions of a state p some transitions of $d_{\mathbb{X}}(p)$. Consider the symbolic transition $\gamma_1 \xrightarrow{a, \bullet}_{\omega} a \triangleright ab.\mathbf{0}$ in Figure 4 (C). The derivation $(a, \bullet, a \triangleright ab.\mathbf{0}) \vdash_{T_{\mathbb{W}}, \mathbb{W}}^a (\varepsilon, \bullet, a \triangleright ab.\mathbf{0}) \vdash_{T_{\mathbb{W}}, \mathbb{W}}^b (b, \bullet, abb \triangleright ab.\mathbf{0})$ means that $a_{\mathbb{W}}(\gamma_1) = a \triangleright a.ab.\mathbf{0} \xrightarrow{\varepsilon, \bullet} a \triangleright ab.\mathbf{0}$ and $ab_{\mathbb{W}}(\gamma_1) = ab \triangleright a.ab.\mathbf{0} \xrightarrow{b, \bullet} abb \triangleright ab.\mathbf{0}$. Note that both the transitions are in the saturated transition system (by soundness of ω and $T_{\mathbb{W}}$). The former is also in the symbolic transition system ω , while the latter is not.

For open nets, take the symbolic transition $l \xrightarrow{\$, \alpha}_{\eta} m$ of $\langle N_4, l \rangle$ in Figure 3. The derivation $(\$, \alpha, m) \vdash_{T_{\mathcal{N}}, \mathbb{N}}^{\$, \alpha} (\$, \alpha, m) \vdash_{T_{\mathcal{N}}, \mathbb{N}}^{\$, \alpha} (\$, \alpha, m\$, \alpha)$ means that $l\$, \alpha \xrightarrow{\$, \alpha} m$ and $l\$, \alpha \xrightarrow{\$, \alpha} m\$, \alpha$. Note that both the transitions are in the saturated transition system (by soundness of η and $T_{\mathcal{N}}$). The former is also in the symbolic transition system η , while the latter is not.

Analogously for $\tau.vy.\bar{y}a + a(b).\bar{a}b_1 \xrightarrow{-|\bar{a}a, \tau}_{\alpha} \bar{a}a$. The derivation $(-|\bar{a}a, \tau, \bar{a}a_1) \vdash_{T_{\mathcal{A}}, \mathbb{A}}^{-|\bar{a}a} (-, \tau, \bar{a}a_1) \vdash_{T_{\mathcal{A}}, \mathbb{A}}^{-|\bar{a}b, \tau} (-|\bar{a}b, \tau, \bar{a}a|\bar{a}b_2)$ means that $\tau.vy.\bar{y}a + a(b).\bar{a}b|\bar{a}a_1 \xrightarrow{-|\bar{a}a, \tau} \bar{a}a_1$ and $\tau.vy.\bar{y}a + a(b).\bar{a}b|\bar{a}a_1 \xrightarrow{-|\bar{a}b, \tau} \bar{a}a|\bar{a}b_2$. Note that both the transitions are in the saturated transition system (by soundness of α and $T_{\mathcal{A}}$). The former is also in the symbolic transition system α , while the latter is not.

Definition 7.2 (Sound Inference System). An inference system T is *sound* w.r.t. a \mathbf{H} -coalgebra $\langle \mathbb{X}, \alpha \rangle$ (or viceversa, $\langle \mathbb{X}, \alpha \rangle$ *satisfies* T) provided that whenever $(c, o, q) \in \alpha(p)$ and $(c, o, q) \vdash_{T, \mathbb{X}}^d (c', o', q')$, then $(c', o', q') \in \alpha(d_{\mathbb{X}}(p))$.

For example, $\langle \mathbb{W}, \alpha_{\mathbb{W}} \rangle$ (i.e., the \mathbf{H} -coalgebra corresponding to the SATTS of swc) satisfies $T_{\mathbb{W}}$, while the coalgebra corresponding to the symbolic transition system ω does not. Analogously for the coalgebra $\langle \mathbb{N}, \alpha_{\mathbb{N}} \rangle$ of open nets and the coalgebra $\langle \mathbb{A}, \alpha_{\mathbb{A}} \rangle$ of asynchronous π -calculus. Hereafter we use $\vdash_{T, \mathbb{X}}$ to mean $\vdash_{T, \mathbb{X}}^{id}$.

Definition 7.3 (Saturated Set). Let \mathbb{X} be a $\Gamma(\mathbf{C})$ -algebra. A set $A \in \mathbf{G}(X)$ is *saturated* in T and \mathbb{X} if it is closed w.r.t. $\vdash_{T, \mathbb{X}}$. The set $\mathbf{S}_{\mathbb{X}}^T(X)$ is the subset of $\mathbf{G}(X)$ containing all and only the saturated sets in T and \mathbb{X} .

Definition 7.4. $\mathbf{S}_T : \mathbf{Alg}_{\Gamma(\mathbf{C})} \rightarrow \mathbf{Alg}_{\Gamma(\mathbf{C})}$ maps each $\mathbb{X} = \langle X, d_{\mathbb{X}}^0, d_{\mathbb{X}}^1, \dots \rangle \in \mathbf{Alg}_{\Gamma(\mathbf{C})}$ into $\mathbf{S}_T(\mathbb{X}) = \langle \mathbf{S}_{\mathbb{X}}^T(X), d_{\mathbf{S}_T(\mathbb{X})}^0, d_{\mathbf{S}_T(\mathbb{X})}^1, \dots \rangle$ where $\forall d \in \|\mathbf{C}\|$, $\forall A \in \mathbf{G}(X)$, $d_{\mathbf{S}_T(\mathbb{X})} A = \{(c_2, o_2, x_2) \text{ s.t. } (c_1, o_1, x_1) \in A \text{ and } (c_1, o_1, x_1) \vdash_{T, \mathbb{X}}^d (c_2, o_2, x_2)\}$. For arrows, it is defined as \mathbf{G} .

There are two differences w.r.t. \mathbf{H} . First, we require that all the sets of transitions are saturated. Then the operators are defined by using the relation $\vdash_{T, \mathbb{X}}^d$.

Notice that \mathbf{S}_T cannot be regarded as a lifting of any endofunctor over $\mathbf{Set}^{|\mathbf{C}|}$. Indeed the definition of $\mathbf{S}_{\mathbb{X}}^T(X)$ depends on the algebraic structure \mathbb{X} . For this reason we cannot use Proposition 5.2.

Now, let $\iota_{\mathbb{X}} : \mathbf{S}_{\mathbb{X}}^T(X) \rightarrow \mathbf{G}(X)$ be the inclusion function. In Appendix D it is proved that it also a $\Gamma(\mathbf{C})$ -homomorphism $\iota_{\mathbb{X}} : \mathbf{S}_T(\mathbb{X}) \rightarrow \mathbf{H}(\mathbb{X})$ and that it extends to a natural transformation.

Lemma 7.5. *Let ι be the family of morphisms $\iota = \{\iota_{\mathbb{X}} : \mathbf{S}_T(\mathbb{X}) \rightarrow \mathbf{H}(\mathbb{X}), \forall \mathbb{X} \in |\mathbf{Alg}_{\Gamma(\mathbf{C})}|\}$. Then $\iota : \mathbf{S}_T \Rightarrow \mathbf{H}$ is a natural transformation.*

It is well-known that every natural transformation between endofunctors induces a functor between the corresponding categories of coalgebras [40]. In our case, $\iota : \mathbf{S}_T \Rightarrow \mathbf{H}$ induces the functor $\mathbf{I} : \mathbf{Coalg}_{\mathbf{S}_T} \rightarrow \mathbf{Coalg}_{\mathbf{H}}$ that maps each \mathbf{S}_T -coalgebra $\alpha : \mathbb{X} \rightarrow \mathbf{S}_T(\mathbb{X})$ into the \mathbf{H} -coalgebra $\alpha; \iota_{\mathbb{X}} : \mathbb{X} \rightarrow \mathbf{H}(\mathbb{X})$.

Let $\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}$ be the full subcategory of $\mathbf{Coalg}_{\mathbf{H}}$ containing the \mathbf{H} -coalgebras $\alpha : \mathbb{X} \rightarrow \mathbf{H}(\mathbb{X})$ that factor through $\iota_{\mathbb{X}}$, i.e., those $\alpha = \alpha'; \iota_{\mathbb{X}}$ for some $\Gamma(\mathbf{C})$ -homomorphisms $\alpha' : \mathbb{X} \rightarrow \mathbf{S}_T(\mathbb{X})$. It is trivial to see that this category is isomorphic to $\mathbf{Coalg}_{\mathbf{S}_T}$.

In order to prove the existence of final object in $\mathbf{Coalg}_{\mathbf{S}_T}$, we show that $\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}$ is the full subcategory of $\mathbf{Coalg}_{\mathbf{H}}$ containing all and only the coalgebras satisfying T . More precisely, we show that $|\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}|$ is a *covariety* of $\mathbf{Coalg}_{\mathbf{H}}$.

Lemma 7.6. *Let $\langle \mathbb{X}, \alpha \rangle$ be a \mathbf{H} -coalgebra. Then it is in $|\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}|$ iff it satisfies T .*

Proposition 7.7. *$|\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}|$ is a covariety of $\mathbf{Coalg}_{\mathbf{H}}$.*

From this follows that we can construct a final object in $\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}$ as the biggest sub-object of $F_{\mathbf{H}}$ satisfying T . Thus the kernel of final morphisms in $\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}$ coincides with the kernel of final morphisms in $\mathbf{Coalg}_{\mathbf{H}}$. This argument extends to $\mathbf{Coalg}_{\mathbf{S}_T}$, since it is isomorphic to $\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}$.

If T is *sound* w.r.t. $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$, then the latter is in $|\mathbf{Coalg}_{\mathbf{H}\mathbf{I}}|$, i.e., $\alpha_{\mathcal{I}} = \alpha'_{\mathcal{I}}; \iota_{\mathbb{X}}$. Note that $\langle \mathbb{X}, \alpha'_{\mathcal{I}} \rangle$ corresponds through the isomorphism to $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ (namely, $\mathbf{I}(\langle \mathbb{X}, \alpha'_{\mathcal{I}} \rangle) = \langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$). Thus, by assuming T to be sound w.r.t. $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$, we have that the kernel of final morphism from $\langle \mathbb{X}, \alpha'_{\mathcal{I}} \rangle$ in $\mathbf{Coalg}_{\mathbf{S}_T}$ coincides with \sim^S .

It is worth to give an intuition about $F_{\mathbf{S}_T}$, the final coalgebra of $\mathbf{Coalg}_{\mathbf{S}_T}$. One can roughly think of $F_{\mathbf{H}}$ (the final coalgebra of $\mathbf{Coalg}_{\mathbf{H}}$) as the standard final coalgebra of

transition systems (with labels in $\|\mathbf{C}\| \times O$), i.e., the coalgebra of all *synchronization trees*. The final coalgebra of $\mathbf{Coalg}_{\mathbf{S}_T}$ is the biggest subcoalgebra of $F_{\mathbf{H}}$ containing all and only those synchronization trees that are sound w.r.t. T . Note that $F_{\mathbf{S}_T}$ is not a “convenient semantics domain” since all the set of transitions of a given state are saturated. In the next subsection, we are going to show the category of normalized coalgebras, where the final coalgebra contains only few “essential” symbolic transitions.

7.2. Normalized Coalgebras. In this subsection we introduce normalized coalgebras, in order to characterize \sim^S without considering the whole SATTS and by relying on the derivation relation $\vdash_{T,\mathbb{X}}$. The following observation is fundamental to explain our idea.

Lemma 7.8. *Let \mathbb{X} be a $\Gamma(\mathbf{C})$ -algebra. For all triples $(c_1, o_1, p_1), (c_2, o_2, p_2) \in \mathbf{H}(\mathbb{X})$, if $(c_1, o_1, p_1) \vdash_{T,\mathbb{X}} (c_2, o_2, p_2)$ then $p_2 = e_{\mathbb{X}}(p_1)$ for some $e \in \|\mathbf{C}\|$. Moreover $\forall q_1 \in \int X$, $(c_1, o_1, q_1) \vdash_{T,\mathbb{X}} (c_2, o_2, e_{\mathbb{X}}(q_1))$.*

Consider a \mathbf{H} -coalgebra $\langle \mathbb{X}, \gamma \rangle$ and the equivalence \sim^γ induced by the final morphism. Suppose that $p \xrightarrow{c_1, o_1}_\gamma p_1$ and $p \xrightarrow{c_2, o_2}_\gamma e_{\mathbb{X}}(p_1)$ such that $(c_1, o_1, p_1) \vdash_{T,\mathbb{X}} (c_2, o_2, e_{\mathbb{X}}(p_1))$. If $\langle \mathbb{X}, \gamma \rangle$ satisfies T (i.e., it is a \mathbf{S}_T -coalgebra), we can forget about the latter transition. Indeed, for all $q \in \int X$, if $q \xrightarrow{c_1, o_1}_\gamma q_1$ then also $q \xrightarrow{c_2, o_2}_\gamma e_{\mathbb{X}}(q_1)$ (since $\langle \mathbb{X}, \gamma \rangle$ satisfies T) and if $p_1 \sim^\gamma q_1$, then also $e_{\mathbb{X}}(p_1) \sim^\gamma e_{\mathbb{X}}(q_1)$ (since \sim^γ is a congruence). Thus, when checking bisimilarity, we can avoid to consider those transitions that are derivable from others. We call such transitions *redundant*.

A wrong way to efficiently characterize \sim^γ by exploiting $\vdash_{T,\mathbb{X}}$, consists in removing all the redundant transitions from $\langle \mathbb{X}, \gamma \rangle$ obtaining a new coalgebra $\langle \mathbb{X}, \beta \rangle$ and then computing \sim^β (i.e., the ordinary bisimilarity on $\langle \mathbb{X}, \beta \rangle$). When considering $\langle \mathbb{X}, \alpha_T \rangle$ (i.e., the \mathbf{H} -coalgebra corresponding to SATTS), this roughly means to build a symbolic transition system and then computing the ordinary bisimilarity over this. But, as we have seen in Section 4, the resulting bisimilarity (denoted by \sim^W) does not coincide with the original one. Generally, this happens when

- (1) $p \xrightarrow{c_1, o_1}_\beta p_1$ and $p \xrightarrow{c_2, o_2}_\beta p_2$ with $(c_1, o_1, p_1) \vdash_{T,\mathbb{X}} (c_2, o_2, e_{\mathbb{X}}(p_1))$ and
- (2) $e_{\mathbb{X}}(p_1) \neq p_2$, but
- (3) $e_{\mathbb{X}}(p_1) \sim^\gamma p_2$.

Notice that $p \xrightarrow{c_2, o_2}_\beta p_2$ is not removed, because it is not considered redundant since $e_{\mathbb{X}}(p_1)$ is different from p_2 (even if semantically equivalent). A transition as the latter is called *semantically redundant* and it causes the mismatch between \sim^β and \sim^γ . Indeed, take a process q that only performs $q \xrightarrow{c_1, o_1}_\beta q_1$ with $p_1 \sim^\gamma q_1$. Clearly $p \not\sim^\beta q$, but $p \sim^\gamma q$. Indeed $(c_1, o_1, q_1) \vdash_{T,\mathbb{X}} (c_2, o_2, e_{\mathbb{X}}(q_1))$ and thus $q \xrightarrow{c_2, o_2}_\beta e_{\mathbb{X}}(q_1)$ (since $\langle \mathbb{X}, \gamma \rangle$ satisfies T) and $p_2 \sim^\gamma e_{\mathbb{X}}(p_1) \sim^\gamma e_{\mathbb{X}}(q_1)$ (since \sim^γ is a congruence).

As an example consider the symbolic transition system of γ_1 (Figure 4(C)). We have that (1) $\gamma_1 \xrightarrow{a, \bullet}_\omega a \triangleright ab.\mathbf{0}$ and $\gamma_1 \xrightarrow{ab, \bullet}_\omega ab \triangleright \varepsilon.\mathbf{0}$ with $(a, \bullet, a \triangleright ab.\mathbf{0}) \vdash_{T_{\mathbb{W}}, \mathbb{W}} (ab, \bullet, ab \triangleright ab.\mathbf{0})$; (2) $ab \triangleright ab.\mathbf{0} \neq ab \triangleright \varepsilon.\mathbf{0}$, but (3) $ab \triangleright ab.\mathbf{0} \sim^S ab \triangleright \varepsilon.\mathbf{0}$. Thus, the symbolic transition $\gamma_1 \xrightarrow{ab, \bullet}_\omega ab \triangleright \varepsilon.\mathbf{0}$ is semantically redundant and it is the reason why $\gamma_2 = \varepsilon \triangleright a.ab.\mathbf{0}$ is not syntactically bisimilar to γ_1 (i.e., $\gamma_1 \not\sim^W \gamma_2$) even if they are saturated bisimilar (as discussed in Section 3).

As a further example consider $\langle N_4, l \rangle$ (Figure 3): (1) $l \xrightarrow{\emptyset, \alpha} q$ and $l \xrightarrow{\mathbb{S}^3, \alpha} m$ with $(\emptyset, \alpha, q) \vdash_{T_{\mathcal{N}, \mathbb{N}}} (\mathbb{S}^3, \alpha, \mathbb{S}_{\mathbb{N}}^3(q))$ and (2) $\mathbb{S}_{\mathbb{N}}^3(q) = q\mathbb{S}^3 \neq m$, but $q\mathbb{S}^3 \sim^S m$. Now consider $\langle N_1, a \rangle$. $a \xrightarrow{\emptyset, \alpha} b$. Clearly $l \not\sim^W a$ but $l \sim^S a$ (as shown in Section 2).

For the asynchronous π -calculus consider the symbolic transitions of $p_1 = \tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$ in Figure 1(B): (1) $p_1 \xrightarrow{-, \tau} \nu y.\bar{y}a_1$ and $p_1 \xrightarrow{-|\bar{a}a, \tau} \bar{a}a_1$ with $(-, \tau, \nu y.\bar{y}a_1) \vdash_{T_{\mathcal{A}, \mathbb{A}}} (-|\bar{a}a, \tau, -|\bar{a}a_{\mathbb{A}}(\nu y.\bar{y}a_1))$; (2) $-|\bar{a}a_{\mathbb{A}}(\nu y.\bar{y}a_1) = \nu y.\bar{y}a|\bar{a}a_1 \neq \bar{a}a_1$, but (3) $\nu y.\bar{y}a|\bar{a}a_1 \sim^S \bar{a}a_1$. Now the process $\tau.\mathbf{0}_1$ only performs $\xrightarrow{-, \tau} \mathbf{0}_1$. Clearly $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1 \not\sim^W \tau.\mathbf{0}_1$, but they are saturated bisimilar (as shown in Section 1).

The above observation tells us that we have to remove not only the redundant transition, i.e., those derivable from $\vdash_{T, \mathbb{X}}$, but also the *semantically redundant* ones. But immediately a problem arises. How can we decide which transitions are semantically redundant, if semantic redundancy itself depends on bisimilarity?

Our solution is the following: we define a category of coalgebras without redundant transitions ($\mathbf{Coalg}_{\mathbf{N}_T}$) and, as a result, a final coalgebra contains no *semantically redundant* transitions.

Definition 7.9 (Normalized Set and Normalization). Let \mathbb{X} be a $\Gamma(\mathbf{C})$ -algebra.

A transition (c', o', q') is *equivalent* to (c, o, q) in T, \mathbb{X} (written $(c', o', q') \equiv_{T, \mathbb{X}} (c, o, q)$) iff $(c', o', q') \vdash_{T, \mathbb{X}} (c, o, q)$ and $(c, o, q) \vdash_{T, \mathbb{X}} (c', o', q')$.

A transition (c', o', q') *dominates* (c, o, q) in T, \mathbb{X} (written $(c', o', q') \prec_{T, \mathbb{X}} (c, o, q)$) iff $(c', o', q') \vdash_{T, \mathbb{X}} (c, o, q)$ and $(c, o, q) \not\vdash_{T, \mathbb{X}} (c', o', q')$.

Let $A \in \mathbf{G}(X)$. A transition $(c, o, q) \in A$ is *redundant in A* w.r.t. T, \mathbb{X} if $\exists (c', o', q') \in A$ such that $(c', o', q') \prec_{T, \mathbb{X}} (c, o, q)$.

The set A is *normalized* w.r.t. T, \mathbb{X} iff it does not contain redundant transitions and it is closed by equivalent transitions. The set $\mathbf{N}_{\mathbb{X}}^T(X)$ is the subset of $\mathbf{G}(X)$ containing all and only the normalized sets w.r.t. T, \mathbb{X} .

The *normalization function* $norm_{T, \mathbb{X}} : \mathbf{G}(X) \rightarrow \mathbf{N}_{\mathbb{X}}^T(X)$ maps $A \in \mathbf{G}(X)$ into $\{(c', o', q') \text{ s.t. } \exists (c, o, q) \in A \text{ s.t. } (c', o', q') \equiv_{T, \mathbb{X}} (c, o, q) \text{ and } (c, o, q) \text{ not redundant in } A \text{ w.r.t. } T, \mathbb{X}\}$.

Recall $\mathcal{W} = \langle \mathbf{Wor}, \mathbb{W}, O_{\mathcal{W}}, tr_{\mathcal{W}} \rangle$ and $T_{\mathcal{W}}$ (introduced in Section 4). Consider the coalgebra $\langle \mathbb{U}, \zeta \rangle$ partially depicted in Figure 8(B). Here we have that $(a, \bullet, \gamma') \vdash_{T_{\mathcal{W}, \mathbb{U}}} (ab, \bullet, b_{\mathbb{U}}(\gamma'))$ but $(ab, \bullet, b_{\mathbb{U}}(\gamma')) \not\vdash_{T_{\mathcal{W}, \mathbb{U}}} (a, \bullet, \gamma')$. Thus $(a, \bullet, \gamma') \prec_{T_{\mathcal{W}, \mathbb{U}}} (ab, \bullet, b_{\mathbb{U}}(\gamma'))$ and then the set $\zeta(\gamma)$, i.e., the set of transitions of γ , is not normalized (w.r.t. $T_{\mathcal{W}, \mathbb{U}}$) since the transition $(ab, \bullet, b_{\mathbb{U}}(\gamma'))$ is redundant in $\zeta(\gamma)$. By applying $norm_{T_{\mathcal{W}, \mathbb{U}}}$ to $\zeta(\gamma)$, we get the normalized set of transitions $\{(a, \bullet, \gamma')\}$ (Figure 8(C)). It is worth noting that in swc, two transitions are equivalent iff they are the same transition.

Now consider $\mathcal{N} = \langle \mathbf{Tok}, \mathbb{N}, \Lambda, tr_{\mathcal{N}} \rangle$, $T_{\mathcal{N}}$ (introduced in Section 4) and the coalgebra $\langle \mathbb{Y}, \gamma \rangle$ partially depicted in Figure 9(A). Here we have that $(\emptyset, \alpha, v) \vdash_{T_{\mathcal{N}, \mathbb{Y}}} (\mathbb{S}^3, \alpha, \mathbb{S}_{\mathbb{Y}}^3(v))$ but $(\mathbb{S}^3, \alpha, \mathbb{S}_{\mathbb{Y}}^3(v)) \not\vdash_{T_{\mathcal{N}, \mathbb{Y}}} (\emptyset, \alpha, v)$. Thus $(\emptyset, \alpha, v) \prec_{T_{\mathcal{N}, \mathbb{Y}}} (\mathbb{S}^3, \alpha, \mathbb{S}_{\mathbb{Y}}^3(v))$ and then the set $\gamma(u)$, i.e., the set of transitions of u , is not normalized (w.r.t. $T_{\mathcal{N}, \mathbb{Y}}$) since the transition $(\mathbb{S}^3, \alpha, \mathbb{S}_{\mathbb{Y}}^3(v))$ is redundant in $\gamma(u)$. By applying $norm_{\mathbb{Y}, T_{\mathcal{N}}}$ to $\gamma(u)$, we get the normalized set of transitions $\{(\emptyset, \alpha, v)\}$ (Figure 9(B)). Also in open Petri nets, two transitions are equivalent iff they are the same transition.

Finally consider $\mathcal{A} = \langle \mathbf{Out}, \mathbb{A}, O_{\mathcal{A}}, tr_{\mathcal{A}} \rangle$, $T_{\mathcal{A}}$ (introduced in Section 4) and the coalgebra $\langle \mathbb{Z}, \delta \rangle$ partially depicted in Figure 9(C). We have that $(-, \tau, q_1) \vdash_{T_{\mathcal{A}, \mathbb{Z}}} (-|\bar{a}a, \tau, -|\bar{a}a_{\mathbb{Z}}(q_1))$ but $(-|\bar{a}a, \tau, -|\bar{a}a_{\mathbb{Z}}(q_1)) \not\vdash_{T_{\mathcal{A}, \mathbb{Z}}} (-, \tau, q_1)$. The same holds for $(-|\bar{a}b, \tau, -|\bar{a}b_{\mathbb{Z}}(q_1))$. Thus

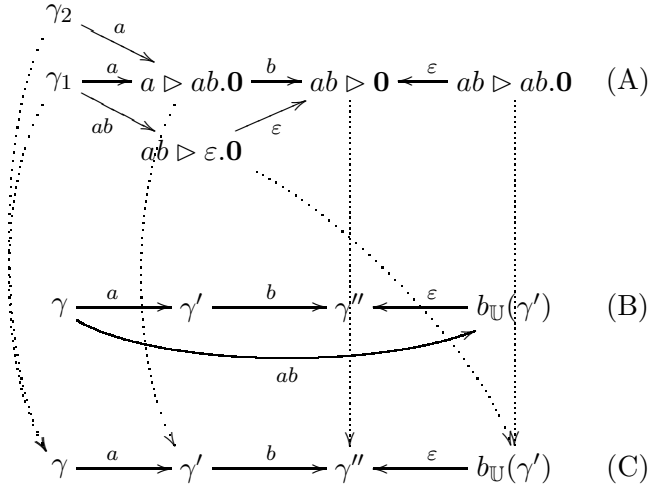


Figure 8: (A) Part of the normalized coalgebra $\langle \mathbb{W}, \alpha_{\mathbb{W}}; norm_{T_{\mathbb{W}}, \mathbb{W}} \rangle$. (B) Part of a not normalized coalgebra $\langle \mathbb{U}, \zeta \rangle$. (C) Part of a normalized coalgebra $\langle \mathbb{U}, \zeta; norm_{T_{\mathbb{W}}, \mathbb{U}} \rangle$. The dotted arrows represent a $\mathbf{N}_{T_{\mathbb{W}}}$ -morphism from $\langle \mathbb{W}, \alpha_{\mathbb{W}}; norm_{T_{\mathbb{W}}, \mathbb{W}} \rangle$ to $\langle \mathbb{U}, \zeta; norm_{T_{\mathbb{W}}, \mathbb{U}} \rangle$.

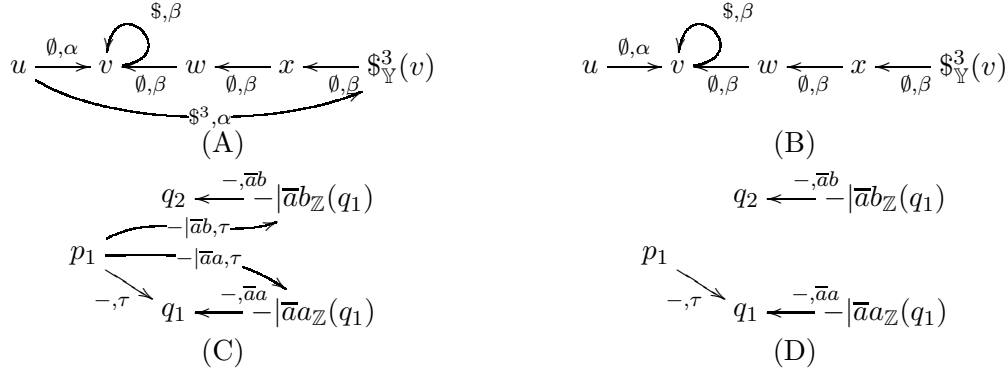


Figure 9: (A) Part of a not normalized coalgebra $\langle \mathbb{Y}, \gamma \rangle$. (B) Part of a normalized coalgebra $\langle \mathbb{Y}, \gamma; norm_{T_{\mathcal{N}}, \mathbb{Y}} \rangle$. (C) Part of a not normalized coalgebra $\langle \mathbb{Z}, \delta \rangle$. (D) Part of a normalized coalgebra $\langle \mathbb{Z}, \delta; norm_{T_{\mathcal{A}}, \mathbb{Z}} \rangle$.

the set $\delta(p_1)$, i.e., the set of transitions of p_1 , is not normalized (w.r.t. $T_{\mathcal{A}}, \mathbb{Z}$) since the transitions $(-\bar{a}a, \tau, -\bar{a}a_{\mathbb{Z}}(q_1))$ and $(-\bar{a}b, \tau, -\bar{a}b_{\mathbb{Z}}(q_1))$ are redundant in $\delta(p_1)$ (they are dominated by $(-\tau, q_1)$). By applying $norm_{\mathbb{Z}, T_{\mathcal{A}}}$ to $\delta(p_1)$, we obtain the normalized set of transitions $\{(-\tau, q_1)\}$ (in Figure 9(D)). Also in the asynchronous π -calculus, two transitions are equivalent iff they are the same transition.

Definition 7.10. $\mathbf{N}_T : \mathbf{Alg}_{\Gamma}(\mathbf{C}) \rightarrow \mathbf{Alg}_{\Gamma}(\mathbf{C})$ maps each $\mathbb{X} = \langle X, d_{\mathbb{X}}^0, d_{\mathbb{X}}^1, \dots \rangle \in \mathbf{Alg}_{\Gamma}(\mathbf{C})$ into $\mathbf{N}_T(\mathbb{X}) = \langle \mathbf{N}_{\mathbb{X}}^T(X), d_{\mathbf{N}_T(\mathbb{X})}^0; norm_{T, \mathbb{X}}, d_{\mathbf{N}_T(\mathbb{X})}^1; norm_{T, \mathbb{X}}, \dots \rangle$. For all $h : \mathbb{X} \rightarrow \mathbb{Y}$, let $\mathbf{H}'(h) : \mathbf{N}_T(\mathbb{X}) \rightarrow \mathbf{H}(\mathbb{Y})$ be the restriction of $\mathbf{H}(h)$ to $\mathbf{N}_T(\mathbb{X})$. Then, $\mathbf{N}_T(h) : \mathbf{N}_T(\mathbb{X}) \rightarrow \mathbf{N}_T(\mathbb{Y})$ is defined as $\mathbf{H}'(h); norm_{T, \mathbb{Y}}$.

Hereafter we will sometimes write $\mathbf{H}(h)$ to mean its restriction $\mathbf{H}'(h)$.

The coalgebra $\langle \mathbb{U}, \zeta \rangle$ (partially depicted in Figure 8) and $\langle \mathbb{Y}, \gamma \rangle$, $\langle \mathbb{Z}, \delta \rangle$ (in Figure 9(A)(C)) are not normalized. In order to get a normalized coalgebra for our running examples, we can normalize their saturated coalgebra $\langle \mathbb{W}, \alpha_{\mathbb{W}} \rangle$, $\langle \mathbb{N}, \alpha_{\mathbb{N}} \rangle$ and $\langle \mathbb{A}, \alpha_{\mathbb{A}} \rangle$ obtaining, respectively, $\langle \mathbb{W}, \alpha_{\mathbb{W}}; norm_{T_{\mathbb{W}}, \mathbb{W}} \rangle$, $\langle \mathbb{N}, \alpha_{\mathbb{N}}; norm_{T_{\mathbb{N}}, \mathbb{N}} \rangle$ and $\langle \mathbb{A}, \alpha_{\mathbb{A}}; norm_{T_{\mathbb{A}}, \mathbb{A}} \rangle$. For γ_1 and γ_2 in Figure 4(C), for the nets in Figure 3 and for the process $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$, this coincides with their SCTS. Section 8 discusses the exact relationship between a SCTS and the transition system that is obtained by normalizing $\alpha_{\mathcal{I}}$.

The most important idea behind normalized coalgebra is in the definition of $\mathbf{N}_{\mathbf{T}}(h)$: we first apply $\mathbf{H}(h)$ and then the normalization $norm_{T, \mathbb{Y}}$. Thus $\mathbf{N}_{\mathbf{T}}$ -morphisms must preserve not all the transitions of the source coalgebras, but only those that are not redundant when mapped into the target.

For instance, consider the function h from $\langle \mathbb{W}, \alpha_{\mathbb{W}}; norm_{T_{\mathbb{W}}, \mathbb{W}} \rangle$ to $\langle \mathbb{U}, \zeta; norm_{T_{\mathbb{W}}, \mathbb{U}} \rangle$ that is partially depicted in Figure 8. Note that the transition $\gamma_1 \xrightarrow{ab, \bullet} ab \triangleright \varepsilon.\mathbf{0}$ is not preserved, but h is however an $\mathbf{N}_{\mathbf{T}}$ -morphisms because the transition $(ab, \bullet, b_{\mathbb{U}}(\gamma'))$ is removed by $norm_{T_{\mathbb{W}}, \mathbb{U}}$. Thus, h forgets about the transition $\gamma_1 \xrightarrow{ab, \bullet} ab \triangleright \varepsilon.\mathbf{0}$ that is indeed semantically redundant.

For the asynchronous π , consider the coalgebra $\langle \mathbb{N}, \alpha_{\mathbb{N}}; norm_{T_{\mathbb{N}}, \mathbb{N}} \rangle$. For the state l , it coincides with the SCTS η (Figure 3(C)). Consider $\langle \mathbb{Y}, \gamma; norm_{T_{\mathbb{N}}, \mathbb{Y}} \rangle$ (partially represented in Figure 9(B)) and the $\Gamma(\mathbf{Tok})$ -homomorphism $h : \mathbb{N} \rightarrow \mathbb{Y}$ that maps l, m, n, o into $u, \$_{\mathbb{Y}}^3(v), x, w$ (respectively) and p, q into v . The morphism is shown in Figure 7. Note that the transition $l \xrightarrow{\$^3, \alpha}_{\eta} m$ is not preserved (i.e., $u \not\xrightarrow{\$^3, \alpha}_{\gamma} h(m)$), but h is however a $\mathbf{N}_{\mathbf{T}}$ -morphism, because the transition $(\$^3, \alpha, h(m)) \in \mathbf{H}(h)(\eta(l))$ is removed by $norm_{T_{\mathbb{N}}, \mathbb{Y}}$. Indeed $h(m) = \$_{\mathbb{Y}}^3(v)$ and $(\emptyset, \alpha, v) \vdash_{T_{\mathbb{N}}, \mathbb{Y}} (\$^3, \alpha, \$_{\mathbb{Y}}^3(v))$. Thus, we forget about $l \xrightarrow{\$^3, \alpha}_{\eta} m$ that is, indeed, semantically redundant.

As a further example, consider the coalgebras $\langle \mathbb{A}, \alpha_{\mathbb{A}}; norm_{T_{\mathbb{A}}, \mathbb{A}} \rangle$. For the state $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$, it coincides with the SCTS α (in Figure 1(B)). Consider $\langle \mathbb{Z}, \delta; norm_{T_{\mathbb{A}}, \mathbb{Z}} \rangle$ (partially represented in Figure 9(D)) and the $\Gamma(\mathbf{Out})$ -homomorphism $h : \mathbb{A} \rightarrow \mathbb{Z}$ shown in Figure 7. Note that for all $i \in \mathcal{N}$ the transitions $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1 \xrightarrow{-|\bar{a}i, \tau}_{\alpha} \bar{a}i$ are not preserved (i.e., $p_1 \not\xrightarrow{-|\bar{a}i, \tau}_{\delta} h(\bar{a}i)$), but h is however a $\mathbf{N}_{\mathbf{T}}$ -morphism, because the transitions $(-|\bar{a}i, \tau, h(\bar{a}i)) \in \mathbf{H}(g)(\alpha(\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1))$ are removed by $norm_{T_{\mathbb{A}}, \mathbb{Z}}$. Indeed $h(\bar{a}i) = -|\bar{a}i_{\mathbb{Z}}(q_1)$ and $(-, \tau, q_1) \vdash_{T_{\mathbb{A}}, \mathbb{Z}} (-|\bar{a}i, \tau, -|\bar{a}i_{\mathbb{Z}}(q_1))$. Thus, we forget about all the transitions $\xrightarrow{-|\bar{a}i, \tau}_{\alpha} \bar{a}i$ that are, indeed, semantically redundant.

7.3. Isomorphism Theorem. Now we prove that $\mathbf{Coalg}_{\mathbf{N}_{\mathbf{T}}}$ is isomorphic to $\mathbf{Coalg}_{\mathbf{S}_{\mathbf{T}}}$.

Definition 7.11 (Saturation). Let \mathbb{X} be a $\Gamma(\mathbf{C})$ -algebra. The saturation function $sat_{T, \mathbb{X}} : \mathbf{G}(X) \rightarrow \mathbf{S}_{\mathbb{X}}^{\mathbf{T}}(X)$ maps all sets of transitions $A \in \mathbf{G}(X)$ into the set $\{(c', o', x') \text{ s.t. } (c, o, x) \in A \text{ and } (c, o, x) \vdash_{T, \mathbb{X}} (c', o', x')\}$.

Saturation is intuitively the opposite of normalization. Indeed saturation adds to a set all the redundant transitions, while normalization junks all of them. Thus, if we take a saturated set of transitions, we first normalize it and then we saturate it, we obtain the original set. Analogously for a normalized set.

However, in order to get such correspondence, we must add a constraint to our theory. Indeed, according to the actual definitions, there could exist a \mathbf{S}_T -coalgebra $\langle \mathbb{X}, \gamma \rangle$ and an infinite descending chain like: $\cdots \prec_{T, \mathbb{X}} p \xrightarrow{c_2, o_2}_\gamma p_2 \prec_{T, \mathbb{X}} p \xrightarrow{c_1, o_1}_\gamma p_1$. In this chain, all the transitions are redundant and thus if we normalize it, we obtain an empty set of transitions.

Definition 7.12 (Normalizable System). A context interactive system $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ is *normalizable w.r.t. T* iff $\forall \mathbb{X} \in \mathbf{Alg}_{\Gamma(\mathbf{C})}$, $\prec_{T, \mathbb{X}}$ is well founded, i.e., there are not infinite descending chains of $\prec_{T, \mathbb{X}}$.

In Appendix A, we show that the context interactive systems for open nets and asynchronous π are normalizable w.r.t. their inference systems.

Lemma 7.13. *Let \mathcal{I} be a normalizable system w.r.t. T . Let \mathbb{X} be $\Gamma(\mathbf{C})$ -algebra and $A \in \mathbf{G}(X)$. Then $\forall (d, o, x) \in A$, either $(d, o, x) \in norm_{T, \mathbb{X}}(A)$ or $\exists (d', o', x') \in norm_{T, \mathbb{X}}(A)$, such that $(d', o', x') \prec_{T, \mathbb{X}} (d, o, x)$.*

The above lemma guarantees that normalizing a set of transitions produces a new set containing all the transitions that are needed to retrieve the original one. Hereafter, we always refer to normalizable systems.

Proposition 7.14. *Let $norm_T$, respectively, sat_T be the families of morphisms $\{norm_{T, \mathbb{X}} : \mathbf{S}_T(\mathbb{X}) \rightarrow \mathbf{N}_T(\mathbb{X}), \forall \mathbb{X} \in |\mathbf{Alg}_{\Gamma(\mathbf{C})}|\}$ and $\{sat_{T, \mathbb{X}} : \mathbf{N}_T(\mathbb{X}) \rightarrow \mathbf{S}_T(\mathbb{X}), \forall \mathbb{X} \in |\mathbf{Alg}_{\Gamma(\mathbf{C})}|\}$. Then $norm_T : \mathbf{S}_T \Rightarrow \mathbf{N}_T$ and $sat_T : \mathbf{N}_T \Rightarrow \mathbf{S}_T$ are natural transformations. More precisely, they are natural isomorphisms, one the inverse of the other.*

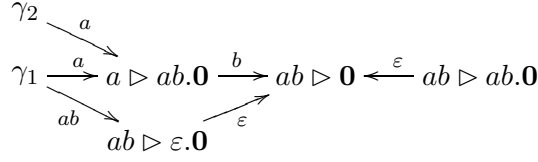
As for the case of the natural transformation ι , we use the fact that that any natural transformation between endofunctors induces a functor between the corresponding categories of coalgebras [40]. In the present case, $norm_T : \mathbf{S}_T \Rightarrow \mathbf{N}_T$ induces the functor $\mathbf{NORM}_T : \mathbf{Coalg}_{\mathbf{S}_T} \rightarrow \mathbf{Coalg}_{\mathbf{N}_T}$ that maps every coalgebra $\langle \mathbb{X}, \alpha \rangle$ in $\langle \mathbb{X}, \alpha; norm_{T, \mathbb{X}} \rangle$ and every cohomomorphism h in itself. Analogously $sat_T : \mathbf{N}_T \Rightarrow \mathbf{S}_T$ induces $\mathbf{SAT}_T : \mathbf{Coalg}_{\mathbf{N}_T} \rightarrow \mathbf{Coalg}_{\mathbf{S}_T}$. These two functors are one the inverse of the other.

Theorem 7.15. *$\mathbf{Coalg}_{\mathbf{S}_T}$ and $\mathbf{Coalg}_{\mathbf{N}_T}$ are isomorphic.*

Thus $\mathbf{Coalg}_{\mathbf{N}_T}$ has a final coalgebra $F_{\mathbf{N}_T}$ and the final morphisms from $\langle \mathbb{X}, \alpha_{\mathcal{I}}; norm_{T, \mathbb{X}} \rangle$ (that is $\mathbf{NORM}_T \langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$) still characterizes \sim^S . This is theoretically very interesting, since the minimal canonical representatives of \sim^S in $\mathbf{Coalg}_{\mathbf{N}_T}$ do not contain any (semantically) redundant transitions and thus they are much smaller than the (possibly infinite) minimal representatives in $\mathbf{Coalg}_{\mathbf{S}_T}$. Pragmatically, it allows for an effective procedure for minimizing that we will discuss in the next section. Notice that minimization is usually unfeasible in $\mathbf{Coalg}_{\mathbf{S}_T}$, since the saturated transitions systems are usually infinite.

8. FROM NORMALIZED COALGEBRAS TO SYMBOLIC MINIMIZATION

In [10], we have introduced a partition refinement algorithm for symbolic bisimilarity. First, it creates a partition P_0 equating all the states (with the same interface) of a symbolic transition system β and then, iteratively, refines this partition by splitting non equivalent states. The algorithm terminates whenever two subsequent partitions are equivalent. It computes the partition P_{n+1} as follows: p and q are equivalent in P_{n+1} iff whenever $p \xrightarrow{c, o}_\beta p_1$ is *not-redundant* in P_n , then $q \xrightarrow{c, o}_\beta q_1$ is *not-redundant* in P_n and p_1, q_1 are equivalent in



$$\begin{aligned}
 P_0 &= \{\gamma_1, \gamma_2, a \triangleright ab.\mathbf{0}, ab \triangleright \mathbf{0}, ab \triangleright \varepsilon.\mathbf{0}, ab \triangleright ab.\mathbf{0}\} \\
 P_1 &= \{\gamma_1, \gamma_2\} \{a \triangleright ab.\mathbf{0}\} \{ab \triangleright \mathbf{0}\} \{ab \triangleright \varepsilon.\mathbf{0}, ab \triangleright ab.\mathbf{0}\} \\
 P_2 &= \{\gamma_1, \gamma_2\} \{a \triangleright ab.\mathbf{0}\} \{ab \triangleright \mathbf{0}\} \{ab \triangleright \varepsilon.\mathbf{0}, ab \triangleright ab.\mathbf{0}\}
 \end{aligned}$$

Figure 10: The partitions computed for $\gamma_1 = \varepsilon \triangleright a.ab.\mathbf{0} + ab.\varepsilon.\mathbf{0}$ and $\gamma_2 = \varepsilon \triangleright a.ab.\mathbf{0}$.

P_n (and viceversa). By “not-redundant in P_n ”, we mean that no transition $p \xrightarrow{c', o'}_{\beta} p'_1$ exists such that $(c', o', p'_1) \vdash_{T, \mathbb{X}} (c, o, p'_2)$ and p'_2, p_1 are equivalent in P_n .

Figure 10 shows the partitions computed by the algorithm for the symbolic transition system ω of $\gamma_1 = \varepsilon \triangleright a.ab.\mathbf{0} + ab.\varepsilon.\mathbf{0}$ and $\gamma_2 = \varepsilon \triangleright a.ab.\mathbf{0}$. In P_0 all the configurations are equivalent since they all have the same interface (more generally, in swc, all the configurations have the same interface). Then in P_1 , $\{ab \triangleright \varepsilon.\mathbf{0}, ab \triangleright ab.\mathbf{0}\}$ are distinguished by all the other configurations because they are the only ones that can perform a transition with ε . Analogously, $ab \triangleright \mathbf{0}$ is different from all the others, because it is the only that performs no transition, while $a \triangleright ab.\mathbf{0}$ is distinguished because it can perform a b transition. Note that γ_1 and γ_2 are equivalent in P_1 , because the transition $\gamma_1 \xrightarrow{ab, \bullet}_{\omega} ab \triangleright \varepsilon.\mathbf{0}$ is redundant in P_0 . Indeed $(a, \bullet, a \triangleright ab.\mathbf{0}) \vdash_{T_{\mathcal{W}}, \mathbb{W}} (ab, \bullet, ab \triangleright ab.\mathbf{0})$ and $ab \triangleright \varepsilon.\mathbf{0}$ is equivalent to $ab \triangleright ab.\mathbf{0}$ in P_0 . The same holds for P_2 .

Figure 11 shows the partitions computed by the algorithm for the SCTS η of the marked nets $\langle N_1, a \rangle$ and $\langle N_4, l \rangle$ of Figure 3. Note that a and l are equivalent in the partition P_1 , because the transition $l \xrightarrow{\$^3, \alpha}_{\eta} m$ is redundant in P_0 . Indeed, $l \xrightarrow{\emptyset, \alpha}_{\eta} q$, $(\emptyset, \alpha, q) \vdash_{T_{\mathcal{N}}, \mathbb{N}} (\$^3, \alpha, q\$^3)$ and m is equivalent to $q\3 in P_0 . Analogously, for the other P_i .

Figure 12 shows the partitions computed by the algorithm for the SCTS α of the asynchronous processes $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$ and $\tau.\mathbf{0}_1$. Since the SCTS of the former process is infinite, our algorithm cannot work in reality. We discuss this issue in the next section and for the time being, we imagine to have a procedure that can manipulate this infinite LTS. First of all, note that all the states with different interfaces are different in P_0 (while in the case of swc and open nets, all the states have the same interface). Moreover, $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$ and $\tau.\mathbf{0}_1$ are equivalent in the partition P_1 , because for all $i \in \omega_0$, the transitions $\tau.\nu y.\bar{y}b + a(b).\bar{a}b_1 \xrightarrow{-|\bar{a}i, \tau}_{\alpha} \bar{a}i$ are redundant in P_0 . Indeed, $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1 \xrightarrow{-\tau}_{\alpha} \nu y.\bar{y}a_1$, $(-, \tau, \nu y.\bar{y}a_1) \vdash_{T_{\mathcal{A}}, \mathbb{A}} (-|\bar{a}i, \tau, \nu y.\bar{y}a|\bar{a}i)$ and $\bar{a}i$ is equivalent to $\nu y.\bar{y}a|\bar{a}i$ in P_0 . Analogously, for P_2 .

The terminal sequence $1 \leftarrow \mathbf{N}_{\mathbf{T}}(1) \leftarrow \mathbf{N}_{\mathbf{T}}^2(1) \leftarrow \dots$ (where 1 is a final $\Gamma(\mathbf{C})$ -algebra) induces a sequence of *approximations* of the final morphism from $\langle \mathbb{X}, \alpha_{\mathcal{I}}; norm_{T, \mathbb{X}} \rangle$ to $F_{\mathbf{N}_{\mathbf{T}}}$. The 0-approximation $!_0 : \mathbb{X} \rightarrow 1$ is the unique morphism in $\mathbf{Alg}_{\Gamma}(\mathbf{C})$. The $n+1$ -approximation $!_{n+1} : \mathbb{X} \rightarrow \mathbf{N}_{\mathbf{T}}^{n+1}(1)$ is defined as $\alpha_{\mathcal{I}}; norm_{T, \mathbb{X}}; \mathbf{N}_{\mathbf{T}}(!_n)$.

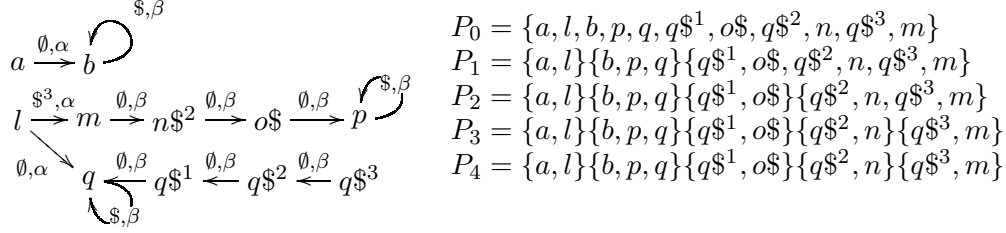


Figure 11: The partitions computed for the marked nets $\langle N_1, a \rangle$ and $\langle N_4, l \rangle$.

$$\begin{array}{ccc}
 \mathbb{X} & \xrightarrow{!_n} & \mathbf{N}_{\mathbf{T}}^n(1) \\
 \downarrow \alpha_{\mathcal{I}; norm_{T, \mathbb{X}}} & \searrow^{!_{n+1}} & \uparrow \\
 \mathbf{N}_{\mathbf{T}}(\mathbb{X}) & \xrightarrow{\mathbf{N}_{\mathbf{T}}(!_n)} & \mathbf{N}_{\mathbf{T}}^{n+1}(1)
 \end{array}$$

In this section, we show that the kernel of the n -approximation $!_n$ coincides with the partition P_n computed by the algorithm. Formally, $!_n(p) = !_n(q)$ iff p is equivalent to q in P_n .

Proposition 8.1. *Let $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ be a context interactive system. Let T and β be, respectively, an inference system and SCTS that are sound and complete for \mathcal{I} . Then $\alpha_{\mathcal{I}}; norm_{T, \mathbb{X}} = \beta; norm_{T, \mathbb{X}}$.*

The above proposition states that the transition systems resulting from the normalization of the saturated $\alpha_{\mathcal{I}}$ coincides with the systems resulting from the normalization of the symbolic β . Note that usually $\beta \neq \beta; norm_{T, \mathbb{X}}$, because our definition of symbolic transition system does not guarantee that β is normalized (according to our definition, also the SATTS is a symbolic transition system). For instance, the symbolic transition system of γ_1 in Figure 4 (C) is normalized, while the one of $\varepsilon \triangleright u.p + v.p$ in Figure 4 (B) is not.

For all the nets in Figure 3, the symbolic transition system is normalized w.r.t. \mathbb{N} but, for the net S_2 in Figure 2, it is not. Indeed both $d \xrightarrow{\theta, \beta} z$ and $d \xrightarrow{y, \beta} zy$, and the former dominates the latter in \mathbb{N} . Also in the case of asynchronous π , the symbolic transition system α is not normalized. Consider the process $\tau.\bar{c}d + a(b).(\bar{c}d|\bar{a}b)_4$. The symbolic transition $\xrightarrow{|\bar{a}b, \tau} \bar{c}d|\bar{a}b_4$ is dominated by $\xrightarrow{|\bar{c}d, \tau} \bar{c}d_4$.

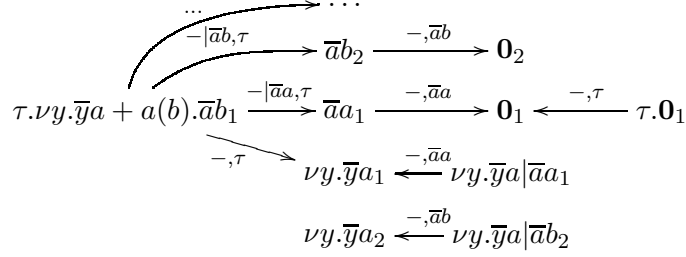
However, when computing the $n + 1$ -approximation $!_{n+1}$, we can simply use β instead of $\beta; norm_{T, \mathbb{X}}$. Indeed,

$$\beta; norm_{T, \mathbb{X}}; \mathbf{N}_{\mathbf{T}}(!_n) = \beta; norm_{T, \mathbb{X}}; \mathbf{H}(!_n); norm_{T, \mathbf{N}_{\mathbf{T}}^n(1)} = \beta; \mathbf{H}(!_n); norm_{T, \mathbf{N}_{\mathbf{T}}^n(1)}$$

where the former equivalence follows from the definition of $\mathbf{N}_{\mathbf{T}}(!_n)$ (Definition 7.10) and the latter follows from Lemma E.3.2 in the Appendix. Thus, $!_{n+1} = \beta; \mathbf{H}(!_n); norm_{T, \mathbf{N}_{\mathbf{T}}^n(1)}$.

Now we can show by induction that $!_{n+1}(p) = !_n(q)$ if and only if p and q belongs to the same partition in P_{n+1} .

The base case trivially holds since $!_0 : \mathbb{X} \rightarrow 1$ maps all the states (with the same interface) into the same element and P_0 equates all the states (with the same interface).



$$\begin{aligned}
P_0 &= \{ \tau.\mathbf{0}_1, \tau.\nu y.\bar{y}a + a(b).\bar{a}b_1, \mathbf{0}_1, \nu y.\bar{y}a_1, \nu y.\bar{y}a|\bar{a}a_1, \bar{a}a_1 \} \\
&\quad \{ \mathbf{0}_2, \nu y.\bar{y}a_2, \nu y.\bar{y}a|\bar{a}b_2, \bar{a}b_2 \} \{ \mathbf{0}_3, \nu y.\bar{y}a_3, \nu y.\bar{y}a|\bar{a}c_3, \bar{a}c_3 \} \dots \\
P_1 &= \{ \tau.\mathbf{0}_1, \tau.\nu y.\bar{y}a + a(b).\bar{a}b_1 \} \{ \mathbf{0}_1, \nu y.\bar{y}a_1 \} \{ \nu y.\bar{y}a|\bar{a}a_1, \bar{a}a_1 \} \\
&\quad \{ \mathbf{0}_2, \nu y.\bar{y}a_2 \} \{ \nu y.\bar{y}a|\bar{a}b_2, \bar{a}b_2 \} \{ \mathbf{0}_3, \nu y.\bar{y}a_3 \} \{ \nu y.\bar{y}a|\bar{a}c_3, \bar{a}c_3 \} \dots \\
P_2 &= \{ \tau.\mathbf{0}_1, \tau.\nu y.\bar{y}a + a(b).\bar{a}b_1 \} \{ \mathbf{0}_1, \nu y.\bar{y}a_1 \} \{ \nu y.\bar{y}a|\bar{a}a_1, \bar{a}a_1 \} \\
&\quad \{ \mathbf{0}_2, \nu y.\bar{y}a_2 \} \{ \nu y.\bar{y}a|\bar{a}b_2, \bar{a}b_2 \} \{ \mathbf{0}_3, \nu y.\bar{y}a_3 \} \{ \nu y.\bar{y}a|\bar{a}c_3, \bar{a}c_3 \} \dots
\end{aligned}$$

Figure 12: The partitions computed for $\tau.\nu y.\bar{y}b + a(b).\bar{a}b$ and $\tau.\mathbf{0}$.

For the inductive case, note that by definition $\beta; \mathbf{H}(!_n)(p)$ is equal to the set of transitions $(c, o, !_n(p_1))$ such that $p \xrightarrow{c, o}_\beta p_1$. Then applying $norm_{T, \mathbf{N}_{\mathbf{T}}^n(1)}$ to this set, means to remove all the transitions $(c, o, !_n(p_1))$ such that there exists a (non equivalent) transition $(c', o', !_n(p'_1)) \in \beta; \mathbf{H}(!_n)(p)$ such that $(c', o', !_n(p'_1)) \vdash_{T, \mathbf{N}_{\mathbf{T}}^n(1)} (c, o, !_n(p_1))$. By Lemma D.2 and Lemma D.3 in the Appendix, the latter becomes: there exists a (non equivalent) transition $(c', o', p'_1) \in \beta(p)$ such that $(c', o', p'_1) \vdash_{T, \mathbb{X}} (c, o, p_2)$ and $!_n(p_1) = !_n(p'_2)$. By inductive hypothesis, $!_n(p_1) = !_n(p'_2)$ iff p_1 and p'_2 belongs to the same partition in P_n . Thus, the normalization $norm_{T, \mathbf{N}_{\mathbf{T}}^n(1)}$ junks away all the redundant transitions in P_n . Summarizing $!_{n+1}(p)$ is equal to the set of transitions $(c, o, !_n(p_1))$ such that $p \xrightarrow{c, o}_\beta p_1$ and the latter is not redundant in P_n . Therefore, $!_{n+1}(p) = !_n(p_1)$, iff whenever $p \xrightarrow{c, o}_\beta p_1$ is *not-redundant* in P_n , then $q \xrightarrow{c, o}_\beta q_1$ is *not-redundant* in P_n and p_1, q_1 are equivalent in P_n .

We end up this section by showing “in algorithmic terms” why normalized coalgebras are not bialgebras. By virtue of Proposition 5.2.2, minimization in bialgebras can be performed, by first forgetting the algebraic structure, and then minimizing in **Set**. Minimization in **Coalg** $_{\mathbf{N}_{\mathbf{T}}}$, instead, heavily relies on the algebraic structure. Indeed in Figure 10, the algorithm needs $ab \triangleright ab.\mathbf{0}$ to compute the partition of γ_1 . Note that γ_1 cannot reach through the symbolic transitions $ab \triangleright ab.\mathbf{0}$, but this is needed for checking if $\gamma_1 \xrightarrow{ab, \bullet}_\omega ab \triangleright \varepsilon.\mathbf{0}$ is redundant. In Figure 11, in order to compute the partitions of l , the algorithm needs the state $q\mathbb{S}^3$ that is not reachable from l . Also in Figure 12, we need the state $\nu y.\bar{y}a|\bar{a}a_1$ that is not reachable from $\tau.\nu y.\bar{y}a + a(b).\bar{a}b_1$.

Summarizing, given a state x of a normalized coalgebra $\langle \mathbb{X}, \beta; norm_{T, \mathbb{X}} \rangle$, in order to compute the partitions on the states reachable from x , the algorithm needs “some” states that are not reachable but that are somehow connect via the algebraic structure \mathbb{X} (such as the states described above). In [10], we have shown that the number of the needed “extra states” is finite in all the interesting cases and it can be computed in the initialization phase of the algorithm. Moreover, it is important to remark here that \mathbb{X} is the only

algebraic structure that is involved in the algorithm: as described above, the normalization $norm_{T, \mathbf{N}_T^n(1)}$ (at iteration $n + 1$) can be computed by just using the algebra \mathbb{X} .

9. CONCLUSIONS AND RELATED WORKS

The paper introduces two coalgebraic models for context interactive systems [8]. In the first one, the saturated transition system is an ordinary structured coalgebra $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ and its final morphism induces \sim^S . The second model is the normalized coalgebra $\langle \mathbb{X}, \alpha_{\mathcal{I}}; norm_{T, \mathbb{X}} \rangle$ that is obtained by pruning all the redundant transitions from the first one. The equivalence induced by its final morphism is still \sim^S , but this characterization is much more convenient. Indeed, in the final normalized coalgebra all the (semantically) redundant transitions are eliminated. Moreover, minimization is usually feasible with normalized coalgebras and coincides with the symbolic minimization algorithm introduced in [10].

As a lateral result, we have obtained coalgebraic models for both open Petri nets and asynchronous π -calculus.

Unfortunately, symbolic minimization is unfeasible in the case of asynchronous π , because the symbolic transition system is infinite. Indeed, in the definition of \sim^1 (Definition 1.2), a process is put in parallel with all possible outputs $\bar{a}b$. Our symbolic transition system eliminates all those outputs whose subjects a are not needed, but yet it considers all the possible objects b . We could have defined a different SCTS that considers only those objects that are strictly needed but, anyway, in the asynchronous π -calculus there are several other sources of infiniteness. Amongst these, one always appears when considering “nominal calculi” where systems are able to generate and communicate names: every time that a system generates a new name and extrudes it, the system goes in a new state that is different from all the previous. HD-Automata [34] are peculiar LTSS that allow to garbage collect names and avoid this further source of infiniteness. As future work, we would like to extend our framework to HD-Automata, so that we will be able to handle systems that generates infinitely many names. In particular we conjecture that the resulting minimization algorithm will generalize both [35] and [38] that provide a partition refinement algorithm for asynchronous and open bisimilarity. The reader is referred to [10] for a more detailed comparison with [35] and [38].

Concerning open bisimilarity [41], a coalgebraic model has been proposed in [22]. However, this is the saturated version, i.e., the one that takes into account all the possible substitutions. In [8], we have given a context interactive system for open π -calculus, and thus our work also provides a coalgebraic model for the “efficient characterization” of open bisimilarity.

Besides open Petri nets, asynchronous and open π -calculus, context interactive systems also generalize *Leifer and Milner’s reactive systems* [29]. The main novelty of our framework consists in having observations and inference rules. The latter generalize the notion of *reactive contexts* of [29]. Indeed c is reactive iff the following inference rule holds.

$$\frac{p \xrightarrow{\tau} q}{c(p) \xrightarrow{\tau} c(q)}$$

Concretely, the main advantage of our framework w.r.t. [29] is that we do not need the existence of RPOs and thus we can avoid those encodings into bigraphs [32] and borrowed contexts [16]. The main disadvantage is that our framework does not provide a constructive definition for the LTS: constructing a sound and complete symbolic transition system is left

to the ingenuity of the researcher. We refer the reader to [8, 4] for a detailed comparison between context interactive systems and reactive systems.

In [6], the first author together with Gadducci and Monreale has shown a reactive system for mobile ambients [12]. Thus, the present work indirectly provides also a coalgebraic semantics for mobile ambients. A coalgebraic model for this calculus has been previously proposed in [23] but it characterizes *action bisimilarity* that is strictly included into *reduction barbed congruence* [30]. Action bisimilarity is defined as the ordinary bisimilarity on the symbolic transition system and thus it is an instance of what we have called syntactic bisimilarity (\sim^W).

Besides their large applicability, normalized coalgebras are interesting for a more theoretical reason: at our knowledge, these are the first example in literature of structured coalgebras that are not bialgebras. Indeed, both the definitions of saturated and normalized set of transitions (Definition 7.3 and 7.9, respectively) strongly rely on the underlying algebraic structures. This is evident in the minimization algorithm in **Coalg_{NT}** that heavily employs the algebraic structure.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous referees (of this and the preliminary version in [9]) for the precious comments that have improved the quality of the paper.

REFERENCES

- [1] Roberto M. Amadio, Ilaria Castellani, and Davide Sangiorgi. On bisimulations for the asynchronous π -calculus. In *Proc. of CONCUR*, volume 1119 of *LNCS*, pages 147–162. Springer, 1996.
- [2] Paolo Baldan, Andrea Corradini, Hartmut Ehrig, Reiko Heckel, and B. König. Bisimilarity and behaviour-preserving reconfiguration of open Petri nets. In *Proc. of CALCO*, volume 4624 of *LNCS*, pages 126–142. Springer, 2007.
- [3] Paolo Baldan, Andrea Corradini, Hartmut Ehrig, and Reiko Heckel. Compositional semantics for open Petri nets based on deterministic processes. *Mathematical Structures in Computer Science*, 15(1):1–35, 2005.
- [4] Filippo Bonchi. *Abstract Semantics by Observable Contexts*. PhD thesis, 2008.
- [5] Filippo Bonchi, Fabio Gadducci, and Barbara König. Process bisimulation via a graphical encoding. In *Proc. of ICGT*, volume 4178 of *LNCS*, pages 168–183, 2006.
- [6] Filippo Bonchi, Fabio Gadducci, and Giacomina Valentina Monreale. Reactive systems, barbed semantics and the mobile ambients. In *Proc. of FOSSACS*, volume 5504 of *LNCS*, pages 272–287. Springer, 2009.
- [7] Filippo Bonchi and Ugo Montanari. Coalgebraic models for reactive systems. In *Proc. of CONCUR*, volume 4703 of *LNCS*, pages 364–379, 2007.
- [8] Filippo Bonchi and Ugo Montanari. Symbolic semantics revisited. In *Proc. of FOSSACS*, volume 4962 of *LNCS*, pages 395–412. Springer, 2008.
- [9] Filippo Bonchi and Ugo Montanari. Coalgebraic symbolic semantics. In *Proc. of CALCO*, volume 5728 of *LNCS*, pages 173–190. Springer, 2009.
- [10] Filippo Bonchi and Ugo Montanari. Minimization algorithm for symbolic bisimilarity. In *Proc. of ESOP*, volume 5502 of *LNCS*, pages 267–284. Springer, 2009.
- [11] Maria Grazia Buscemi and Ugo Montanari. Cc-pi: A constraint-based language for specifying service level agreements. In *Proc. of ESOP*, volume 4421 of *LNCS*, pages 18–32. Springer, 2007.
- [12] Luca Cardelli and Andrew D. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000.
- [13] Andrea Corradini, Martin Große-Rhode, and Reiko Heckel. Structured transition systems as lax coalgebras. *Elect. Notes in Theor. Comp. Sci.*, 11, 1998.

- [14] Andrea Corradini, Martin Große-Rhode, and Reiko Heckel. A coalgebraic presentation of structured transition systems. *Theoretical Computer Science*, 260:27–55, 2001.
- [15] Andrea Corradini, Reiko Heckel, and Ugo Montanari. Tile transition systems as structured coalgebras. In *Proc. of FCT*, pages 13–38, 1999.
- [16] Hartmut Ehrig and Barbara König. Deriving bisimulation congruences in the DPO approach to graph rewriting. In *Proc. of FOSSACS*, volume 2987 of *LNCS*, pages 151–166. Springer, 2004.
- [17] Marcelo P. Fiore, Eugenio Moggi, and Davide Sangiorgi. A fully abstract model for the π -calculus. *Information and Computation*, 179(1):76–117, 2002.
- [18] Marcelo P. Fiore and Daniel Turi. Semantics of name and value passing. In *LICS*, pages 93–104. IEEE, 2001.
- [19] Fabio Gadducci. Term graph rewriting and the π -calculus. *Programming Languages and Semantics*, volume 2895 of *LNCS*, pages 37–54. Springer, 2003.
- [20] Fabio Gadducci and Ugo Montanari. A concurrent graph semantics for mobile ambients. *Mathematical Foundations of Programming Semantics*, volume 45 of *Elect. Notes in Theor. Comp. Sci.* Elsevier Science, 2001.
- [21] Fabio Gadducci and Ugo Montanari. The tile model. *Proof, Language and Interaction: Essays in honour of Robin Milner*. MIT Press, 1999.
- [22] Neil Ghani, Kidane Yemane, and Björn Victor. Relationally staged computations in calculi of mobile processes. *Elect. Notes in Theor. Comp. Sci.*, 106:105–120, 2004.
- [23] Daniel Hausmann, Till Mossakowski, and Lutz Schröder. A coalgebraic approach to the semantics of the ambient calculus. *Theoretical Computer Science*, 366(1-2):121–143, 2006.
- [24] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138(2):353–389, 1995.
- [25] Kohei Honda and Mario Tokoro. An object calculus for asynchronous communication. In *Proc. of ECOOP*, volume 512 of *LNCS*, pages 133–147. Springer, 1991.
- [26] Paris C. Kanellakis and Scott A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. *Information and Computation*, 86(1):43–68, 1990.
- [27] Ekkart Kindler. A compositional partial order semantics for Petri net components. In *Proc. of ATPN*, volume 1248 of *LNCS*, pages 235–252. Springer, 1997.
- [28] Alexander Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, 2000.
- [29] James J. Leifer and Robin Milner. Deriving bisimulation congruences for reactive systems. In *Proc. of CONCUR*, volume 1877 of *LNCS*, pages 243–258. Springer, 2000.
- [30] Massimo Merro and Francesco Zappa Nardelli. Bisimulation proof methods for mobile ambients. In *Proc. of ICALP*, volume 2719 of *LNCS*, pages 584–598. Springer, 2003.
- [31] Robin Milner. *Communicating and Mobile Systems: the π -Calculus*. Cambridge University Press, 1999.
- [32] Robin Milner. Bigraphical reactive systems. In *Proc. of CONCUR*, volume 2154 of *LNCS*, pages 16–35. Springer, 2001.
- [33] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, i and ii. *Information and Computation*, 100(1):1–40, 41–77, 1992.
- [34] Ugo Montanari and Marco Pistore. An introduction to history dependent automata. *Elect. Notes in Theor. Comp. Sci.*, 10, 1997.
- [35] Ugo Montanari and Marco Pistore. Finite state verification for the asynchronous π -calculus. In *Proc. of TACAS*, volume 1579 of *LNCS*, pages 255–269. Springer, 1999.
- [36] Ugo Montanari and Vladimiro Sassone. Dynamic congruence vs. progressing bisimulation for ccs. *Fundam. Inform.*, 16(1):171–199, 1992.
- [37] Joachim Parrow and Björn Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *LICS*, pages 176–185. IEEE, 1998.
- [38] Marco Pistore and Davide Sangiorgi. A partition refinement algorithm for the π -calculus. *Information and Computation*, 164(2):264–321, 2001.
- [39] Wolfgang Reisig. *Petri Nets: An Introduction*. EATCS Monographs on Theoretical Computer Science. Springer Verlag, 1985.
- [40] Jan Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.
- [41] Davide Sangiorgi. A theory of bisimulation for the π -calculus. *Acta Inf.*, 33(1):69–97, 1996.
- [42] Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *LICS*, pages 280–291. IEEE, 1997.

- [43] Lucian Wischik and Philippa Gardner. Strong bisimulation for the explicit fusion calculus. In *Proc. of FOSSACS*, volume 2987 of *LNCS*, pages 484–498. Springer, 2004.
- [44] Lucian Wischik and Philippa Gardner. Explicit fusions. *Theoretical Computer Science*, 340(3):606–630, 2005.

APPENDIX A. NORMALIZABLE SYSTEMS

In this appendix we show that the context interactive system $\mathcal{N} = \langle \mathbf{Tok}, \mathbb{N}, \Lambda, tr_{\mathcal{N}} \rangle$ is normalizable w.r.t. $T_{\mathcal{N}}$ and that $\mathcal{A} = \langle \mathbf{Out}, \mathbb{A}, O_{\mathcal{A}}, tr_{\mathcal{A}} \rangle$ is normalizable w.r.t. $T_{\mathcal{A}}$ (all these are defined in Section 4). Then we show an example of a not normalizable systems.

Proposition A.1. $\mathcal{N} = \langle \mathbf{Tok}, \mathbb{N}, \Lambda, tr_{\mathcal{N}} \rangle$ is normalizable w.r.t. $T_{\mathcal{N}}$.

Proof. Recall that arrows of \mathbf{Tok} are multisets (over sets of input places) and that $c; d = e$ if and only if $c \oplus d = e$. Then, for all $\Gamma(\mathbf{Tok})$ -algebra \mathbb{Y} ,

$$(c_1, \lambda, x_1) \prec_{T_{\mathcal{N}}, \mathbb{Y}} (c_2, \lambda', x_2)$$

only if the multiset c_1 is strictly included into the multiset c_2 . Since all multisets are finite also the descending chains must be finite. \square

Proposition A.2. $\mathcal{A} = \langle \mathbf{Out}, \mathbb{A}, O_{\mathcal{A}}, tr_{\mathcal{A}} \rangle$ is normalizable w.r.t. $T_{\mathcal{A}}$.

Proof. Recall that arrows of \mathbf{Out} are contexts representing parallel output processes and that $c; d = e$ if and only if e is the syntactic composition of c with d . Then, for all $\Gamma(\mathbf{Out})$ -algebra \mathbb{Y} ,

$$(c_1, o, q_1) \prec_{T_{\mathcal{A}}, \mathbb{Y}} (c_2, o', q_2)$$

only if the context c_2 is the parallel composition of c_1 with some other outputs. Since all contexts are finite then the descending chains must be also finite. \square

Example A.3. As an example of not normalizable context interactive system consider the category \mathbf{NAT}_{\geq} defined as follow:

- objects are natural numbers and ∞ ,
- there is an arrow $n \rightarrow m$, if $n \geq m$ or if $n = \infty$.

Since for any two objects n, m there is only one arrow, we call this arrow just as $n \rightarrow m$. Consider now a context interactive system $\mathcal{NAT}_{\geq} = \langle \mathbf{NAT}_{\geq}, \mathbb{X}, O, tr \rangle$ for some \mathbb{X}, O, tr . Let T be the tile system that states that all contexts preserve transitions.

We have that \mathcal{NAT}_{\geq} is not normalizable with respect to T . Indeed, let \mathbb{F} be the final $\Gamma(\mathbf{NAT}_{\geq})$. In this algebra there is only one element \star for each sort (natural number), and an arrow $n \rightarrow m$ of \mathbf{NAT}_{\geq} is interpreted in the function mapping \star of sort n into \star of sort m . Since $\infty \rightarrow n$ can be decomposed in $\infty \rightarrow n + 1 \rightarrow n$, then

$$(\infty \rightarrow n + 1, l, \star) \prec_{T, \mathbb{F}} (\infty \rightarrow n, l, \star).$$

This trivially leads to an infinite descending chain.

APPENDIX B. PROOFS OF SECTION 4

Proposition 4.3. \sim^S is the coarsest bisimulation congruence.

Proof. Since \sim^S is a saturated bisimulation, then it is also a congruence: if $p \sim^S q$, then for all contexts $c_{\mathbb{X}}$, it holds that $c_{\mathbb{X}}(p) \sim^S c_{\mathbb{X}}(q)$.

In order to prove that it is the coarsest bisimulation congruence, we prove that any bisimulation congruence R is a saturated bisimulation.

Suppose that $p R q$. Suppose that $c_{\mathbb{X}}(p) \xrightarrow{o} p'$. Since R is a congruence, then $c_{\mathbb{X}}(p) R c_{\mathbb{X}}(q)$. Since R is a bisimulation $c_{\mathbb{X}}(q) \xrightarrow{o} q'$ and $p' R q'$. Thus R is a saturated bisimulation. \square

Proposition 4.4. Let $\langle N_1, m_1 \rangle$ and $\langle N_2, m_2 \rangle$ be two marked nets both with interface I . Thus $\langle N_1, m_1 \rangle \sim^N \langle N_2, m_2 \rangle$ iff $\langle N_1, m_1 \rangle \sim_I^S \langle N_2, m_2 \rangle$.

Proof. The definition of \sim^S instantiated to the context interactive system \mathcal{N} , requires that $\langle N_1, m_1 \rangle$ and $\langle N_2, m_2 \rangle$ (a) make the same transitions with the rule (TR) and (b) they are still equivalent when adding multisets $i \in I^\oplus$. The definition of \sim^N instead requires that the two nets perform the same transitions with both the rule (TR) and the rule (IN). But the latter rule just adds multisets $i \in I^\oplus$ and thus it is just the same of point (b) above. \square

Proposition 4.5. Let p, q be asynchronous π -processes, and let $n \geq \max \text{fn}(p \cup q)$. Then $p \sim^1 q$ iff $p_n \sim_n^S q_n$.

Proof. Let $R = \{(p, q) \mid p_n \sim_n^S q_n \ n \geq \max \text{fn}(p \cup q)\}$. In order to prove that $p_n \sim_n^S q_n$ implies $p \sim^1 q$, we prove that R is an 1-bisimulation, i.e., an $\sigma\tau$ -bisimulation closed under composition with output processes. Suppose that $p \xrightarrow{\bar{i}(j)} p'$ (the cases of τ and output are easier). First of all observe that $p_n \sim_n^S q_n$ implies that $\forall m \geq n \ p_m \sim_m^S q_m$. Now since j is fresh, we have that $j-1 \geq n$, and thus $p_{j-1} \sim_{j-1}^S q_{j-1}$. By definition of $tr_{\mathcal{A}}$, we have that $p_{j-1} \xrightarrow{\bar{i}(j)}_{\mathcal{A}} p'_j$ and, since $p_{j-1} \sim_{j-1}^S q_{j-1}$, it follows that $q_{j-1} \xrightarrow{\bar{i}(j)}_{\mathcal{A}} q'_j$ and $p'_j \sim_j^S q'_j$ and then, $p' R q'$. Again by definition of $tr_{\mathcal{A}}$, we have that $q \xrightarrow{\bar{i}(j)} q'$. This prove that R is an $\sigma\tau$ -bisimulation. Now we have to prove that it is closed under composition with output processes, but this is immediate since \sim^S is a congruence w.r.t. composition with output processes.

Let R be the ω -sorted relation, such that $\forall n \in \omega, R_n = \{(p_n, q_n) \mid p \sim^1 q, \ n \geq \max \text{fn}(p \cup q)\}$. In order to prove that $p \sim^1 q$ implies $p_n \sim_n^S q_n$, we prove that R is a saturated bisimulation. Let $c \in \mathbf{Out}[n, m]$ and suppose that $c(p_n)_m \xrightarrow{\bar{i}(j)}_{\mathcal{A}} p'_{m+1}$ (the case of τ and output are easier). By definition of $tr_{\mathcal{A}}$, $c(p) \xrightarrow{\bar{i}(m+1)} p'$. Now, since $p \sim^1 q$, by definition of 1-bisimulation, it follows that $c(p) \sim^1 c(q)$ because contexts c are just parallel output processes. Thus $c(q) \xrightarrow{\bar{i}(m+1)} q'$ and $p' \sim^1 q'$. By definition of $tr_{\mathcal{A}}$, it follows that $c(q_n) \xrightarrow{\bar{i}(j)}_{\mathcal{A}} q'_{m+1}$ and, by definition of R , that $p'_{m+1} R_{m+1} q'_{m+1}$. \square

Theorem 4.12. Let \mathcal{I} be a context interactive system, β a context transition system and T an inference system. If β and T are sound and complete w.r.t. \mathcal{I} , then $\sim^{SYM} = \sim^S$.

Proof. Let $R = \{R_i \subseteq X_i \times X_i \mid i \in |\mathbf{C}|\}$ be the $|\mathbf{C}|$ -sorted family of relations, such that $\forall j \in |\mathbf{C}|$,

$$R_j = \{(c(p_i), c(q_i)) \mid c \in \mathbf{C}[i, j], p_i \sim_i^{SYM} q_i\}.$$

In order to prove that $\sim^{SYM} \subseteq \sim^S$ we prove that R is a saturated bisimulation. Suppose that $a_j R_j b_j$ thus there exists $c \in \mathbf{C}[i, j]$ such that $c(p_i) = a_j$, $c(q_i) = b_j$ and $p_i \sim_i^{SYM} q_i$.

Hereafter, in order to make lighter the notation, we avoid to specify the sort of processes and contexts. Thus, p, q, a, b stand for, respectively, p_i, q_i, a_j, b_j .

Suppose that $d(a) = d(c(p)) \xrightarrow{l_1} p_1$ then, by definition of SATTS, $p \xrightarrow{c;d,l_1}_S p_1$. By completeness of β and T , we have that $p \xrightarrow{c_2,l_2}_\beta p_2$ such that $p \xrightarrow{c_2,l_2}_\beta p_2 \vdash_T p \xrightarrow{c;d,l_1}_S p_1$, i.e., $\exists e, e_1 \in \|\mathbf{C}\|$ such that:

- $e \xrightarrow[l_1]{l_2} e_1 \in \Phi(T)$,
- $c_2; e = c; d$,
- $e_1(p_2) = p_1$.

Since $p \sim^{SYM} q$, $q \xrightarrow{c_3,l_3}_\beta q_3 \vdash_T q \xrightarrow{c_2,l_2} q_2$ and $p_2 \sim^{SYM} q_2$. From the former we have that $\exists f, f_1 \in \|\mathbf{C}\|$ such that:

- $f \xrightarrow[l_2]{l_3} f_1 \in \Phi(T)$,
- $c_3; f = c_2$,
- $f_1(q_3) = q_2$.

Since $\Phi(T)$ is closed by composition, then $f; e \xrightarrow[l_1]{l_3} f_1; e_1 \in \Phi(T)$. Moreover $c_3; f; e = c_2; e = c; d$. Thus $q \xrightarrow{c_3,l_3}_\beta q_3 \vdash_T q \xrightarrow{c;d,l_1}_S e_1(q_2)$. Since β and T are sound, it follows that $q \xrightarrow{c;d,l_1}_S e_1(q_2)$, i.e., $d(b) \xrightarrow{l_1} e_1(q_2)$. Since $p_2 \sim^{SYM} q_2$, then $e_1(p_2) R e_1(q_2)$, i.e., $p_1 R e_1(q_2)$.

For proving that $\sim^S \subseteq \sim^{SYM}$, take $p \sim^S q$: if $p \xrightarrow{c,l}_\beta p_1$ then also $p \xrightarrow{c,l}_S p_1$ and, since $p \sim^S q$, $q \xrightarrow{c,l}_S q_1$ with $p_1 \sim^S q_1$. By completeness of β , we have that $q \xrightarrow{c_1,l_1}_\beta q_1 \vdash_T q \xrightarrow{c,l}_S q_1$. \square

Proposition 4.13. ω and $T_{\mathcal{W}}$ are sound and complete w.r.t. \mathcal{W} .

Proof. Proving soundness is quite easy. Just observe that (1) if $u \triangleright p \xrightarrow{v,\bullet}_\omega u' \triangleright p'$ then $uv \triangleright p \xrightarrow{\bullet} u' \triangleright p'$ and (2) the ‘‘monotonicity property’’ described in Section 3 holds.

For proving completeness we suppose that $w_{\mathbb{W}}(\gamma) \xrightarrow{\bullet} \gamma''$ (i.e., $\gamma \xrightarrow{w,\bullet}_S \gamma''$) and we proceed by induction on the structure of the process of the configuration γ . The inductive case for $\gamma = u \triangleright p_1 + p_2$ is trivial. The base case is as follows.

Take $\gamma = u_1 \triangleright u_2.p$ (thus $w_{\mathbb{W}}(\gamma) = u_1 w \triangleright u_2.p$). Note that γ'' must be equal to $u_1 w \triangleright p$ (by the rules defining $tr_{\mathcal{W}}$). Since the configuration $u_1 w \triangleright u_2.p$ perform a transition then u_2 is a prefix of $u_1 w$ (again by the rules defining $tr_{\mathcal{W}}$). There are two possible cases: either u_2 is a prefix of u_1 or not.

In the former case, by the leftmost rule (defining ω), $\gamma = u_1 \triangleright u_2.p \xrightarrow{\varepsilon,\bullet}_\omega u_1 \triangleright p$ and by definition of $\vdash_{T_{\mathcal{W}}}$, this transition derives $\gamma \xrightarrow{w,\bullet}_S w_{\mathbb{W}}(u_1 \triangleright p) = u_1 w \triangleright p = \gamma''$.

If u_2 is not a prefix of u_1 , then there exists $u, v \in A^*$ such that $w = vu$ and $u_2 = u_1.v$. By the central rule (defining ω), we have that $\gamma = u_1 \triangleright u_1.v.p \xrightarrow{v,\bullet}_\omega u_1.v \triangleright p$ and, by definition of $\vdash_{T_{\mathcal{W}}}$, this transition derives $\gamma \xrightarrow{vu,\bullet}_S w_{\mathbb{W}}(u_1.v \triangleright p) = u_1 w \triangleright p = \gamma''$. \square

Proposition 4.15. *Let $\langle N_1, m_1 \rangle$ and $\langle N_2, m_2 \rangle$ be two marked nets both with interface I . Thus $\langle N_1, m_1 \rangle \sim^{NS} \langle N_2, m_2 \rangle$ iff $\langle N_1, m_1 \rangle \sim_I^{SYM} \langle N_2, m_2 \rangle$.*

Proof. The general condition of symbolic bisimilarity

• if $p \xrightarrow{c, \circ}_\beta p'$, then $q \xrightarrow{c_1, \circ_1}_\beta q'_1$ and $q \xrightarrow{c_1, \circ_1}_\beta q'_1 \vdash_T q \xrightarrow{c, \circ} q'$ and $p' R_k q'$.

for the context interactive system \mathcal{N} , the SCTS η and the inference system $T_{\mathcal{N}}$, becomes

• if $\langle N_1, m_1 \rangle \xrightarrow{i, \lambda}_\eta \langle N_1, m'_1 \rangle$, then $\langle N_2, m_2 \rangle \xrightarrow{j, \lambda_1}_\eta \langle N_2, m'_2 \rangle$ and $\langle N_2, m_2 \rangle \xrightarrow{j, \lambda_1}_\eta \langle N_2, m'_2 \rangle \vdash_{T_{\mathcal{N}}} \langle N_2, m_2 \rangle \xrightarrow{i, \lambda}_\eta \langle N_2, m''_2 \rangle$ and $\langle N_1, m'_1 \rangle R_I \langle N_2, m''_2 \rangle$.

From the latter, we have that $\lambda_1 = \lambda$ and there exists $k \in I^\oplus$ such that $i = j \oplus k$ and $m''_2 = m'_2 \oplus k$. These are the conditions of net-symbolic bisimilarity. \square

Proposition 4.16. *η and $T_{\mathcal{N}}$ are sound and complete w.r.t. \mathcal{N} .*

Proof. We have to prove:

• (completeness) if $\langle N, m \rangle \xrightarrow{i, \lambda}_S \langle N, m' \rangle$ then

$$\langle N, m \rangle \xrightarrow{i_1, \lambda_1}_\eta \langle N, m_1 \rangle \quad \text{and} \quad \langle N, m \rangle \xrightarrow{i_1, \lambda_1}_\eta \langle N, m_1 \rangle \vdash_{T_{\mathcal{N}}} \langle N, m \rangle \xrightarrow{i, \lambda} \langle N, m' \rangle.$$

• (soundness) if $\langle N, m \rangle \xrightarrow{i_1, \lambda_1}_\eta \langle N, m_1 \rangle$ and $\langle N, m \rangle \xrightarrow{i_1, \lambda_1}_\eta \langle N, m_1 \rangle \vdash_{T_{\mathcal{N}}} \langle N, m \rangle \xrightarrow{i, \lambda} \langle N, m' \rangle$ then $\langle N, m \rangle \xrightarrow{i, \lambda}_S \langle N, m' \rangle$.

Let us prove completeness. If $N, m \oplus i \xrightarrow{\lambda}_{\mathcal{N}} N, m'$, then there exists a transition $t \in T$, such that $\lambda(t) = l$ and $m \oplus i = \bullet t \oplus c$ and $m' = t \bullet \oplus c$. We can take $c_1 = m \ominus (m \cap \bullet t)$ and $i_1 = \bullet t \ominus (m \cap \bullet t)$. and apply the only rule of η , and $N, m \xrightarrow{i_1, \lambda}_\eta N, t \bullet \oplus c_1$. Note that $i_1 \subseteq i$, since by definition i_1 is the smallest multiset that allow the transition t . Thus let $x = i \ominus i_1$. We have $N, m \xrightarrow{i_1, \lambda}_\eta N, t \bullet \oplus c_1 \vdash_{T_{\mathcal{N}}} N, m \xrightarrow{i, \lambda}_{T_{\mathcal{N}}(\eta)} N, m'$. Indeed:

- $i_1 \oplus x = i$;
- $t \bullet \oplus c_1 \oplus x = m'$, because $c_1 \oplus x = m \ominus (\bullet t \cap m) \oplus x = m \oplus \bullet t \ominus (\bullet t \cap m) \oplus x \ominus \bullet t = m \oplus i_1 \oplus x \ominus \bullet t = m \oplus i \ominus \bullet t = c$.

For proving soundness observe that if $\langle N, m \rangle \xrightarrow{i_1, \lambda_1}_\eta \langle N, m_1 \rangle$ then $\langle N, m \oplus i_1 \rangle \xrightarrow{\lambda_1} \langle N, m_1 \rangle$. Moreover, if $\langle N, m \rangle \xrightarrow{i_1, \lambda_1}_\eta \langle N, m_1 \rangle \vdash_{T_{\mathcal{N}}} \langle N, m \rangle \xrightarrow{i, \lambda} \langle N, m' \rangle$, then $\lambda_1 = \lambda$ there exists $x \in I^\oplus$ such that $i_1 \oplus x = i$ and $m_1 \oplus x = m'$.

Thus, $\langle N, m \oplus i_1 \oplus x \rangle \xrightarrow{\lambda_1} \langle N, m_1 \oplus x \rangle$, that means $\langle N, m \rangle \xrightarrow{i, \lambda}_S \langle N, m' \rangle$. \square

Proposition 4.18. *Let p, q be asynchronous π -processes, and let $n \geq \max \text{fn}(p \cup q)$. Then $p \sim^a q$ iff $p_n \sim_n^{SYM} q_n$.*

Proof. Here we prove that if $p_n \sim_n^{SYM} q_n$ then $p \sim^a q$ (the other implication is analogous).

Let $R = \{p, q \mid p_n \sim_n^{SYM} q_n\}$ be a symmetric relation. We prove that R is an asynchronous bisimulation.

Take $p_n \sim_n^{SYM} q_n$ and suppose that $p \xrightarrow{\bar{i}(j)} p'$ and j is fresh. First, observe that $\forall m \geq n$, $p_m \sim_m^{SYM} q_m$. Then, note that since j is fresh, $j - 1 \geq n$ and thus $p_{j-1} \sim_{j-1}^{SYM} q_{j-1}$.

By definition of $tr_{\mathcal{A}}$, $p_{j-1} \xrightarrow{-, \bar{i}()}_\alpha p'_j$. Now since $p_{j-1} \sim_{j-1}^{SYM} q_{j-1}$, q_{j-1} must answer with a transition $q_{j-1} \xrightarrow{c, \circ}_\alpha q''$ such that $q_{j-1} \xrightarrow{c, \circ}_\alpha q'' \vdash_{T_{\mathcal{A}}} q_{j-1} \xrightarrow{-, \bar{i}()}_\alpha q'_j$ and $p'_j \sim_j^{SYM} q'_j$. By definition of $T_{\mathcal{A}}$, the only such transition is $q_{j-1} \xrightarrow{-, \bar{i}()}_\alpha q'_j$. Now, by definition of α , we have

that $q \xrightarrow{\bar{i}(j)} q'$ and, by definition of R , $p'Rq'$. We can proceed analogously in the case of output and τ .

For the input, suppose that $p \xrightarrow{i(j)} p'$. Then $p_n \xrightarrow{-|\bar{i}j, \tau}_\alpha p'_{n'}$ where $n' = \max\{j, n\}$. Now since $p_n \sim^{SYM} q_n$, q_n must answer with a transition $q_n \xrightarrow{c, \sigma}_\alpha q''$ such that $q_n \xrightarrow{c, \sigma}_\alpha q'' \vdash_{T_A}$ $q_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q'_{n'}$ and $p'_{n'} \sim_{n'}^{SYM} q'_{n'}$.

By definition of T_A there are two possibilities:

- $q_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q'_{n'}$ and $p'_{n'} \sim_{n'}^{SYM} q'_{n'}$. Thus $q \xrightarrow{i(j)} q'$ and $p'Rq'$.
- $q_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q''_{n'}$ and by using the rule $(TAU)_{-|\bar{i}j}$, $q_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q''_{n'} \vdash_{T_A} q_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q''_{n'}|\bar{i}j$ and $p'_{n'} \sim_{n'}^{SYM} (q''_{n'}|\bar{i}j)_{n'}$. Thus $q \xrightarrow{\tau} q''$ and $p'Rq''|\bar{i}j$.

Note that it is correct to write $q'_{n'}$, since $\text{fn}(q') = \text{fn}(q|\bar{i}j) \subseteq n'$. The same holds also for $(q''|\bar{i}j)_{n'}$: $\text{fn}(q'') = \text{fn}(q) \subseteq n$ and thus $\text{fn}(q''|\bar{i}j) \subseteq n'$. \square

Proposition 4.19. α and T_A are sound and complete w.r.t. \mathcal{A} .

Proof. We have to prove:

- (completeness) if $p_n \xrightarrow{c, \mu}_S q_m$ then $p_n \xrightarrow{c, \mu'}_\alpha q'_{m'}$ and $p_n \xrightarrow{c, \mu'}_\alpha q'_{m'} \vdash_{T_A} p_n \xrightarrow{c, \mu}_S q_m$.
- (soundness) if $p_n \xrightarrow{c, \mu'}_\alpha q'_{m'}$ and $p_n \xrightarrow{c, \mu'}_\alpha q'_{m'} \vdash_{T_A} p_n \xrightarrow{c, \mu}_S q_m$ then $p_n \xrightarrow{c, \mu}_S q_m$.

For soundness just observe that if $p \xrightarrow{c, \mu}_\alpha p'$ then $c(p) \xrightarrow{\mu} p'$ and that all the rules of T_A are sound. Let us prove completeness. Suppose that $\mu = \tau$ (the other cases are easier): $p_n \xrightarrow{c, \tau}_S q_m$ implies that $c(p_n)_m \xrightarrow{\tau} q_m$ and $c \in \mathbf{Out}[n, m]$. By definition of tr_A , it follows that $c(p) \xrightarrow{\tau} q$. Since c could be only the parallel composition of outputs, by the definition of the operational semantics of asynchronous π , it follows that either $p \xrightarrow{\tau} q'$ (such that $q = c(q')$) or $c \xrightarrow{\bar{i}j} c'$ (where $c = -|\bar{i}j|c'$) and $p \xrightarrow{i(j)} q'$ (such that $q = c'|q'$).

In the former case, by definition of α , we have that $p_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q'_n$ and using the rule $(TAU)_c$ of T_A , we have that $p_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q'_n \vdash_{T_A} p_n \xrightarrow{c, \tau}_\alpha c(q'_n)_m = q_m$. In the latter case, by definition of α , we have that $p_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q'_{n'}$ where $n' = \max\{j, n\}$. Now, take $c' \in \mathbf{Out}[n', m]$, by the rule $(TAU)_{c'}$ of T_A , we have that $p_n \xrightarrow{-|\bar{i}j, \tau}_\alpha q'_{n'} \vdash_{T_A} p_n \xrightarrow{-|\bar{i}j|c', \tau}_\alpha c'(q'_{n'})_m = q_m$. \square

APPENDIX C. PROOFS OF SECTION 6

Theorem 6.3. Let $\mathcal{I} = \langle \mathbf{C}, \mathbb{X}, O, tr \rangle$ be a context interactive system. Then $\langle \mathbb{X}, \alpha_{\mathcal{I}} \rangle$ is a \mathbf{H} -coalgebra.

Proof. We have to prove that $\alpha_{\mathcal{I}} : \mathbb{X} \rightarrow \mathbf{H}(\mathbb{X})$ is a $\Gamma(\mathbf{C})$ -homomorphism, i.e., that $\forall x \in X$ and $\forall d \in \Gamma(\mathbf{C})$, $\alpha_{\mathcal{I}}(d_{\mathbb{X}}(x)) = d_{\mathbf{H}(\mathbb{X})}(\alpha_{\mathcal{I}}(x))$.

Let $(c, l, y) \in \alpha_{\mathcal{I}}(d_{\mathbb{X}}(x))$ be a saturated transition of $d_{\mathbb{X}}(x)$. Then by definition of $\alpha_{\mathcal{I}}$, $(d; c, l, y) \in \alpha_{\mathcal{I}}(x)$. By definition of $d_{\mathbf{H}(\mathbb{X})}$ and by $(d; c, l, y) \in \alpha_{\mathcal{I}}(x)$, follows that $(c, l, y) \in d_{\mathbf{H}(\mathbb{X})}(\alpha_{\mathcal{I}}(x))$.

Now let $(c, l, y) \in d_{\mathbf{H}(\mathbb{X})}(\alpha_{\mathcal{I}}(x))$. By definition of $d_{\mathbf{H}(\mathbb{X})}$ we have that $(d; c, l, y) \in \alpha_{\mathcal{I}}(x)$ and, analogously to before, $(c, l, y) \in \alpha_{\mathcal{I}}(d_{\mathbb{X}}(x))$. \square

APPENDIX D. PROOFS OF SECTION 7.1

Before proving Lemma 7.6 and Proposition 7.7, we prove some important lemmas about the derivation relation $\vdash_{T,\mathbb{X}}^d$. Moreover, at the end of this appendix we formally show the existence of the final object in $\mathbf{Coalg}_{\mathbf{S}_T}$.

Lemma D.1 (composition of $\vdash_{T,\mathbb{X}}^d$).

If $(c, l, x) \vdash_{T,\mathbb{X}}^d (c', l', x') \vdash_{T,\mathbb{X}}^e (c'', l'', x'')$ then $(c, l, x) \vdash_{T,\mathbb{X}}^{d;e} (c'', l'', x'')$.

Proof. From the hypothesis we derives that there exists $d', d'', e', e'' \in \|\mathbf{C}\|$ such that $d; c' = c; d'$ and $d' \xrightarrow{l} d''$ and $e; c'' = c'; e'$ and $e' \xrightarrow{l'} e''$ such that $d''_{\mathbb{X}}(x) = x'$ and $e''_{\mathbb{X}}(x') = x''$.

From all this, we derive that $(d; e); c'' = c; (d'; e')$ and that $d'; e' \xrightarrow{l} d''; e''$ and that $e''_{\mathbb{X}}(d''_{\mathbb{X}}(x)) = x''$. Then the thesis immediately follows. \square

Lemma D.2 ($\vdash_{T,\mathbb{X}}^d$ is preserved by homomorphisms).

Let $h : \mathbb{X} \rightarrow \mathbb{Y}$ be a $\Gamma(\mathbf{C})$ -homomorphism. If $(c, l, x) \vdash_{T,\mathbb{X}}^d (c', l', x')$, then $(c, l, h(x)) \vdash_{T,\mathbb{Y}}^d (c', l', h(x'))$.

Proof. If $(c, l, x) \vdash_{T,\mathbb{X}}^d (c', l', x')$, then there exists $d' \in \|\mathbf{C}\|$ such that $d; c' = c; d'$ and $d' \xrightarrow{l} d''$ and $d''_{\mathbb{X}}(x) = x'$. Since h is an homomorphism $h(x') = h(d''_{\mathbb{X}}(x)) = d''_{\mathbb{Y}}(h(x))$, and then $(c, l, h(x)) \vdash_{T,\mathbb{Y}}^d (c', l', h(x'))$. \square

Lemma D.3 ($\vdash_{T,\mathbb{X}}^d$ is reflected by homomorphisms).

Let $h : \mathbb{X} \rightarrow \mathbb{Y}$ be a $\Gamma(\mathbf{C})$ -homomorphism. If $(c, l, h(x)) \vdash_{T,\mathbb{Y}}^d (c', l', y')$, then $\exists x' \in \mathbb{X}$, such that $h(x') = y'$ and $(c, l, x) \vdash_{T,\mathbb{X}}^d (c', l', x')$.

Proof. From the hypothesis we derive that there exists $f \in \|\mathbf{C}\|$ such that $c; f = d; c'$ and $f \xrightarrow{l} f'$ and $f'_{\mathbb{Y}}(h(x)) = y'$. Since h is an homomorphism, $h(f'_{\mathbb{X}}(x)) = y'$. Then we have that $(c, l, x) \vdash_{T,\mathbb{X}}^d (c', l', f'_{\mathbb{X}}(x))$. \square

Proposition D.4. $\mathbf{S}_T : \mathbf{Alg}_{\Gamma(\mathbf{C})} \rightarrow \mathbf{Alg}_{\Gamma(\mathbf{C})}$ is a functor.

Proof. First of all, we have to show that $\forall \mathbb{X} \in |\mathbf{Alg}_{\Gamma(\mathbf{C})}|, \mathbf{S}_T(\mathbb{X}) \in |\mathbf{Alg}_{\Gamma(\mathbf{C})}|$. Notice that all the operators $d_{\mathbf{S}_T(\mathbb{X})}$ are well defined, i.e., $\forall A \in \mathbf{S}_{\mathbb{X}}^T(X), d_{\mathbf{S}_T(\mathbb{X})}(A)$ is still a saturated set of transitions, i.e., it is closed w.r.t. $\vdash_{T,\mathbb{X}}^{id}$. Then we have to prove that $id_{\mathbf{S}_T(\mathbb{X})}$ coincides with the identity function. This is trivial since $id_{\mathbf{S}_T(\mathbb{X})}(A)$ consists in closing the set of transition A w.r.t. $\vdash_{T,\mathbb{X}}^{id}$. But since A is saturated, it is already closed. Finally we have to prove that $(c; d)_{\mathbf{S}_T(\mathbb{X})} = c_{\mathbf{S}_T(\mathbb{X})}; d_{\mathbf{S}_T(\mathbb{X})}$, but this is trivial consequence of Lemma D.1.

Then we have to prove that if $h : \mathbb{X} \rightarrow \mathbb{Y}$ in $\mathbf{Alg}_{\Gamma(\mathbf{C})}$, then also $\mathbf{S}_T(h) : \mathbf{S}_T(\mathbb{X}) \rightarrow \mathbf{S}_T(\mathbb{Y})$. This follows easily by Lemma D.2 and Lemma D.3.

Then preservation of identity and arrow composition follows from the fact that \mathbf{S}_T is defined as \mathbf{H} on arrows and on the fact that \mathbf{H} is a functor. \square

Lemma D.5. The inclusion $\iota_{\mathbb{X}} : \mathbf{S}_{\mathbb{X}}^T(X) \rightarrow \mathbf{G}(X)$ is a $\Gamma(\mathbf{C})$ -homomorphism from the algebra $\mathbf{S}_T(\mathbb{X})$ to $\mathbf{H}(\mathbb{X})$.

Proof. We have to prove that for all $A \in \mathbf{S}_{\mathbb{X}}^{\mathbf{T}}(X)$ and $d \in \|\mathbf{C}\|$, $\iota_{\mathbb{X}}(d_{\mathbf{S}_{\mathbf{T}}(\mathbb{X})}(A)) = d_{\mathbf{H}(\mathbb{X})}(\iota(A))$.

Let $(c, l, x) \in \iota_{\mathbb{X}}(d_{\mathbf{S}_{\mathbf{T}}(\mathbb{X})}(A))$, then there exists $(c', l', x') \in A$ such that $(c', l', x') \vdash_{T, \mathbb{X}}^d (c, l, x)$. By definition of $\vdash_{T, \mathbb{X}}^d$, we also have that $(c', l', x') \vdash_{T, \mathbb{X}} (d; c, l, x)$ and since A is saturated, then $(d; c, l, x) \in A$. Since $\iota_{\mathbb{X}}$ is simply the inclusion, we also have that $(d; c, l, x) \in \iota_{\mathbb{X}}(A)$ and thus, by definition of $d_{\mathbf{H}(\mathbb{X})}$, $(c, l, x) \in d_{\mathbf{H}(\mathbb{X})}(\iota_{\mathbb{X}}(A))$.

The other direction is analogous. \square

Lemma 7.5. *Let ι be the family of morphisms $\iota = \{\iota_{\mathbb{X}} : \mathbf{S}_{\mathbf{T}}(\mathbb{X}) \rightarrow \mathbf{H}(\mathbb{X}), \forall \mathbb{X} \in |\mathbf{Alg}_{\Gamma}(\mathbf{C})|\}$. Then $\iota : \mathbf{S}_{\mathbf{T}} \Rightarrow \mathbf{H}$ is a natural transformation.*

Proof. From Lemma D.5, it follows that each $\iota_{\mathbb{X}}$ is a morphism in $\mathbf{Alg}_{\Gamma}(\mathbf{C})$. The fact that $\forall h : \mathbb{X} \rightarrow \mathbb{Y}, \iota_{\mathbb{Y}}; \mathbf{H}(h) = \mathbf{S}_{\mathbf{T}}(h); \iota_{\mathbb{X}}$ follows from the fact that, by definition, $\mathbf{S}_{\mathbf{T}}(h) = \mathbf{H}(h)$. \square

Lemma 7.6. *Let $\langle \mathbb{X}, \alpha \rangle$ be a \mathbf{H} -coalgebra. Then it is in $|\mathbf{Coalg}_{\mathbf{H}^{\mathbf{T}}}|$ iff it satisfies T .*

Proof. Let $\langle \mathbb{X}, \alpha \rangle$ be a \mathbf{H} -coalgebra. If it satisfies T , then $\forall x \in \mathbb{X}, \alpha(x) \in \mathbf{S}_{\mathbb{X}}^{\mathbf{T}}(X)$. This means that α factor through the inclusion $\iota_{\mathbb{X}} : \mathbf{S}_{\mathbf{T}}(\mathbb{X}) \rightarrow \mathbf{H}(\mathbb{X})$.

If T is not sound, then $\exists x \in \mathbb{X}, d \in \|\mathbf{C}\|$ such that $(c, l, y) \in \alpha(x)$ and $(c, l, y) \vdash_{T, \mathbb{X}}^d (c', l', y')$ and $(c', l', y') \notin \alpha(d_{\mathbb{X}}(x)) = d_{\mathbf{H}(\mathbb{X})}(\alpha(x))$. From $(c, l, y) \vdash_{T, \mathbb{X}}^d (c', l', y')$, we have that $(c, l, y) \vdash_{T, \mathbb{X}} (d; c', l', y')$. From this setting follows that $(d; c', l', y') \notin \alpha(c)$ because, otherwise, by definition of $d_{\mathbf{H}(\mathbb{X})}$, we would have that $(c', l', y') \in d_{\mathbf{H}(\mathbb{X})}(\alpha(x))$. Thus $\alpha(x)$ is not saturated, i.e., $\alpha(x) \notin \mathbf{S}_{\mathbb{X}}^{\mathbf{T}}(X)$. \square

Proposition 7.7. $|\mathbf{Coalg}_{\mathbf{H}^{\mathbf{T}}}|$ is a covariety of $\mathbf{Coalg}_{\mathbf{H}}$, i.e., is closed under:

- (1) subcoalgebras,
- (2) homomorphic images,
- (3) sums.

Proof. A coalgebra $\langle \mathbb{X}, \alpha \rangle$ is a subcoalgebra of $\langle \mathbb{Y}, \beta \rangle$ if there is an arrow $m : \langle \mathbb{X}, \alpha \rangle \rightarrow \langle \mathbb{Y}, \beta \rangle$ that is mono in all its components (for a more formal definition look at Appendix F).

The fact that $|\mathbf{Coalg}_{\mathbf{H}^{\mathbf{T}}}|$ is closed under subcoalgebras means that whenever there is a subcoalgebra $m : \langle \mathbb{X}, \alpha \rangle \rightarrow \langle \mathbb{Y}, \beta \rangle$ in $\mathbf{Coalg}_{\mathbf{H}}$ such that $\langle \mathbb{Y}, \beta \rangle \in |\mathbf{Coalg}_{\mathbf{H}^{\mathbf{T}}}|$, then also $\langle \mathbb{X}, \alpha \rangle \in |\mathbf{Coalg}_{\mathbf{H}^{\mathbf{T}}}|$. This can be easily proved by employing Lemma 7.6.

If $\langle \mathbb{Y}, \beta \rangle \in |\mathbf{Coalg}_{\mathbf{H}^{\mathbf{T}}}|$, then it satisfies T . Suppose ab absurdum that $\langle \mathbb{X}, \alpha \rangle$ does not satisfy T . Then there exists $x \in |\mathbb{X}|$, $(c_1, l_1, x_1) \in \alpha(x)$ and $(c_2, l_2, x_2) \notin \alpha(x)$ such that $(c_1, l_1, x_1) \vdash_{\mathbb{X}, T} (c_2, l_2, x_2)$. Now, since m is a cohomomorphism we have that $(c_1, l_1, m(x_1)) \in \beta(m(x))$. By Lemma D.2, it follows that $(c_1, l_1, m(x_1)) \vdash_{\mathbb{Y}, T} (c_2, l_2, m(x_2))$. Since $\langle \mathbb{Y}, \beta \rangle$ satisfies T then also $(c_2, l_2, m(x_2)) \in \beta(m(x))$. At this point, since m is a cohomomorphism then it must exist a $x_3 \in \mathbb{X}$, such that $(c_1, l_1, x_3) \in \alpha(x)$ and $m(x_3) = m(x_2)$. But since m is mono in all its components, then $x_2 = x_3$ and thus $(c_1, l_1, x_2) \in \alpha(x)$ against the hypothesis.

Let $h : \langle \mathbb{X}, \alpha \rangle \rightarrow \langle \mathbb{Y}, \beta \rangle$ be an arrow in $\mathbf{Coalg}_{\mathbf{H}}$. The homomorphic image of $\langle \mathbb{X}, \alpha \rangle$ through h , is the coalgebra $\langle \mathbb{I}, \gamma \rangle$ induced by the unique factorization of $h = e; m$ (as shown below), where e is an arrow with all components epi and m is an arrow with all components mono (look at Appendix F).

$$\begin{array}{ccc}
\mathbb{X} & \xrightarrow{h} & \mathbb{Y} \\
\downarrow \alpha & \searrow e & \downarrow \beta \\
& \mathbb{I} & \\
& \downarrow \text{---} & \\
\mathbf{H}(\mathbb{X}) & \xrightarrow{\mathbf{H}(h)} & \mathbf{H}(\mathbb{Y}) \\
\downarrow \mathbf{H}(e) & & \downarrow \mathbf{H}(m) \\
& \mathbf{H}(\mathbb{I}) &
\end{array}$$

The fact that $|\mathbf{Coalg}_{\mathbf{H}^I}|$ is closed under homomorphic images means that whenever there is a cohomomorphism $h : \langle \mathbb{X}, \alpha \rangle \rightarrow \langle \mathbb{Y}, \beta \rangle$ in $\mathbf{Coalg}_{\mathbf{H}}$ such that $\langle \mathbb{X}, \alpha \rangle \in |\mathbf{Coalg}_{\mathbf{H}^I}|$, then also $\langle \mathbb{I}, \gamma \rangle \in |\mathbf{Coalg}_{\mathbf{H}^I}|$. This can be easily proved by employing Lemma 7.6.

If $\langle \mathbb{X}, \alpha \rangle \in |\mathbf{Coalg}_{\mathbf{H}^I}|$, then it satisfies T . Suppose ab absurdo that $\langle \mathbb{I}, \gamma \rangle$ does not satisfy T . Then there exists an $i \in \mathbb{I}$, $(c_1, l_1, i_1) \in \gamma(i)$ and $(c_2, l_2, i_2) \notin \gamma(i)$ such that $(c_1, l_1, i_1) \vdash_{\mathbb{I}, T} (c_2, l_2, i_2)$. Now, since e is epi in all its components, there exists x_1 , such that $e(x_1) = i_1$ and since e is a cohomomorphism there exists $x \in \mathbb{X}$ such that $h(x) = i$ and $(c_1, l_1, x_1) \in \alpha(x)$. By Lemma D.3 and by $(c_1, l_1, i_1) \vdash_{\mathbb{I}, T} (c_2, l_2, i_2)$, it follows that there exists $x_2 \in \mathbb{X}$ such that $e(x_2) = i_2$ and $(c_1, l_1, x_1) \vdash_{\mathbb{X}, T} (c_2, l_2, x_2)$. Now, since $\langle \mathbb{X}, \alpha \rangle$ satisfies T , then also $(c_2, l_2, x_2) \in \alpha(x)$. And now, since e is a cohomomorphism $(c_2, l_2, i_2) \in \gamma(i)$ against the initial hypothesis.

In $\mathbf{Coalg}_{\mathbf{H}}$, all the colimits are defined as in $\mathbf{Alg}_{\Gamma(\mathbf{C})}$ (for classical argument in coalgebra theory). Recalling that $\mathbf{Alg}_{\Gamma(\mathbf{C})}$ is isomorphic to $\mathbf{Set}^{\mathbf{C}}$, it is easy to see that all colimits exists and they are constructed as in \mathbf{Set} . Thus, it is trivial to prove that if $\langle \mathbb{X}, \alpha \rangle$ and $\langle \mathbb{Y}, \beta \rangle$ satisfy T , also their sum, i.e., $\langle \mathbb{X} + \mathbb{Y}, \alpha + \beta \rangle$, satisfies T . \square

Theorem D.6. $\mathbf{Coalg}_{\mathbf{H}^I}$ has final object $F_{\mathbf{H}^I}$.

Proof. The proof is a standard argument in the theory of coalgebras.

Hereafter, we write “ \mathbf{H}^I -coalgebra” as a short-hand for “ \mathbf{H} -coalgebra in $|\mathbf{Coalg}_{\mathbf{H}^I}|$ ”. In order to construct $F_{\mathbf{H}^I}$, consider all the unique \mathbf{H} -cohomorphisms of \mathbf{H}^I -coalgebras to $F_{\mathbf{H}}$ (the final object of $\mathbf{Coalg}_{\mathbf{H}}$). Consider their homomorphic images through these final morphisms. All of them are subobjects of $F_{\mathbf{H}}$ and all of them are \mathbf{H}^I -coalgebras, because $|\mathbf{Coalg}_{\mathbf{H}^I}|$ is closed under homomorphic images. Now, since these are subobjects of $F_{\mathbf{H}}$, we can define $F_{\mathbf{H}^I}$ as their union. In order to prove that $F_{\mathbf{H}^I}$ is final, it is important to note that it is still a subcoalgebra of $F_{\mathbf{H}}$ (Corollary 1.4.14 of [28]), and thus we have a mono $m : F_{\mathbf{H}^I} \rightarrow F_{\mathbf{H}}$ ³. Then for any \mathbf{H}^I -coalgebra $\langle \mathbb{X}, \alpha \rangle$ there exists a morphism to $F_{\mathbf{H}^I}$ since it is the union of all the images to $F_{\mathbf{H}}$. Then, this morphism is unique since m is mono. Moreover, $F_{\mathbf{H}^I}$ satisfies T , since covarieties are also closed by unions of subcoalgebras.

Another way of proving this theorem relies on Corollary 2.2.4 of [28]. From such corollary and from Proposition 7.7, it follows that $\mathbf{Coalg}_{\mathbf{H}^I}$ is a reflective subcategory of $\mathbf{Coalg}_{\mathbf{H}}$. \square

Corollary D.7. $\mathbf{Coalg}_{\mathbf{S}_T}$ has final object $F_{\mathbf{S}_T}$.

Proof. From the above theorem and from the fact that $\mathbf{Coalg}_{\mathbf{S}_T}$ is isomorphic to $\mathbf{Coalg}_{\mathbf{H}^I}$. \square

³For this is important to notice that all morphisms in $M_{\mathbf{C}}$ (defined in Appendix F) are also mono.

Corollary D.8. *Let $\langle \mathbb{X}, \alpha \rangle$ be a \mathbf{S}_T -coalgebras. Let $!_{\langle \mathbb{X}, \alpha \rangle}^{\mathbf{H}}$ be the unique morphism to $F_{\mathbf{H}}$ and let $!_{\langle \mathbb{X}, \alpha \rangle}^{\mathbf{S}_T}$ be the unique morphism to $F_{\mathbf{S}_T}$. Thus*

$$!_{\langle \mathbb{X}, \alpha \rangle}^{\mathbf{H}}(x) = !_{\langle \mathbb{X}, \alpha \rangle}^{\mathbf{H}}(y) \text{ if and only if } !_{\langle \mathbb{X}, \alpha \rangle}^{\mathbf{S}_T}(x) = !_{\langle \mathbb{X}, \alpha \rangle}^{\mathbf{S}_T}(y).$$

Proof. Note that $F_{\mathbf{H}I} = \mathbf{I}(F_{\mathbf{S}_T})$ for \mathbf{I} being the functor described in Section 7.1. Moreover, from the proof of the above theorem, we have that $F_{\mathbf{H}I}$ is a subobject of $F_{\mathbf{H}}$. \square

APPENDIX E. PROOFS OF SECTION 7.2

In this appendix we prove several lemmas that describe interesting properties of the normalization function. In particular these properties are useful to show that \mathbf{N}_T is a functor. Hereafter, we will always implicitly assume to have a normalizable context interactive system (Definition 7.12).

Lemma 7.8. *Let \mathbb{X} be a $\Gamma(\mathbf{C})$ -algebra. If $(c_1, o_1, p_1) \vdash_{T, \mathbb{X}} (c_2, o_2, p_2)$ then $p_2 = e_{\mathbb{X}}(p_1)$ for some $e \in \|\mathbf{C}\|$. Moreover $\forall q_1 \in Y, (c_1, o_1, q_1) \vdash_{T, \mathbb{X}} (c_2, o_2, e_{\mathbb{X}}(q_1))$.*

Proof. Both observations trivially follows from the definition of $\vdash_{T, \mathbb{X}}$ (Def. 7.1). \square

Lemma 7.13. *Let \mathcal{I} be a normalizable system w.r.t. T . Let \mathbb{X} be $\Gamma(\mathbf{C})$ -algebra and $A \in \mathbf{G}(X)$. Then $\forall (d, o, x) \in A$, either $(d, o, x) \in \text{norm}_{T, \mathbb{X}}(A)$ or $\exists (d', o', x') \in \text{norm}_{T, \mathbb{X}}(A)$, such that $(d', o', x') \prec_{T, \mathbb{X}} (d, o, x)$.*

Proof. If there exists no (d', o', x') with $(d', o', x') \prec_{T, \mathbb{X}} (d, o, x)$, then $(d, o, x) \in \text{norm}_{T, \mathbb{X}}(A)$. If it exists, then consider a chain $\cdots \prec_{T, \mathbb{X}} (d_2, l_2, x_2) \prec_{T, \mathbb{X}} (d^1, l_1, x_1) \prec_{T, \mathbb{X}} (d, l, x)$. Since $\prec_{T, \mathbb{X}}$ is well founded there exists no infinite chains like this. Let $(d', l', x') \in A$ be the last element of such a chain. Since it is the last, it is not redundant and then $(d', l', x') \in \text{norm}_{T, \mathbb{X}}(A)$. Moreover since $\prec_{T, \mathbb{X}}$ is transitive (as proved in the next lemma), we have that $(d', l', x') \prec_{T, \mathbb{X}} (d, l, x)$. \square

Lemma E.1. *Let \mathcal{I} be a context interactive system and T be an inference system. Let \mathbb{X}, \mathbb{Y} be $\Gamma(\mathbf{C})$ -algebras.*

- (1) $\prec_{T, \mathbb{X}}$ is transitive,
(or better, if $(d'', l'', x'') \vdash_{T, \mathbb{X}} (d', l', x') \prec_{T, \mathbb{X}} (d, l, x)$ then $(d'', l'', x'') \prec_{T, \mathbb{X}} (d, l, x)$),
- (2) If $(d'_0, l'_0, x'_0) \equiv_{T, \mathbb{X}} (d_0, l_0, x_0) \prec_{T, \mathbb{X}} (d_1, l_1, x_1) \equiv_{T, \mathbb{X}} (d'_1, l'_1, x'_1)$ then
 $(d'_0, l'_0, x'_0) \prec_{T, \mathbb{X}} (d'_1, l'_1, x'_1)$,
- (3) If $h : \mathbb{X} \rightarrow \mathbb{Y}$ and $(d, l, x) \equiv_{T, \mathbb{X}} (d', l', x')$ then $(d, l, h(x)) \equiv_{T, \mathbb{Y}} (d', l', h(x'))$.

Proof. Suppose that $(d'', l'', x'') \vdash_{T, \mathbb{X}} (d', l', x') \prec_{T, \mathbb{X}} (d, l, x)$, then we have both

$$(d'', l'', x'') \vdash_{T, \mathbb{X}} (d', l', x') \vdash_{T, \mathbb{X}} (d, l, x) \text{ and } (d, l, x) \not\vdash_{T, \mathbb{X}} (d', l', x').$$

We derive $(d'', l'', x'') \vdash_{T, \mathbb{X}} (d, l, x)$ by the former, and $(d, l, x) \not\vdash_{T, \mathbb{X}} (d'', l'', x'')$ by the latter (otherwise if $(d, l, x) \vdash_{T, \mathbb{X}} (d'', l'', x'')$ then also $(d, l, x) \vdash_{T, \mathbb{X}} (d', l', x')$).

For the second point is sufficient to note that

$$(d'_0, l'_0, x'_0) \vdash_{T, \mathbb{X}} (d_0, l_0, x_0) \vdash_{T, \mathbb{X}} (d_1, l_1, x_1) \vdash_{T, \mathbb{X}} (d'_1, l'_1, x'_1),$$

and then $(d'_0, l'_0, x'_0) \vdash_{T, \mathbb{X}} (d'_1, l'_1, x'_1)$. Moreover $(d'_1, l'_1, x'_1) \not\vdash_{T, \mathbb{X}} (d'_0, l'_0, x'_0)$, since otherwise $(d_1, l_1, x_1) \vdash_{T, \mathbb{X}} (d_0, l_0, x_0)$.

For the third point we use that $\vdash_{T, \mathbb{X}}$ is preserved by homomorphisms (Lemma D.2). \square

Lemma E.2. *If $(d, l, x) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$, then $(d, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}(A)$.*

Proof. If $(d, l, x) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$, then by definition of $c_{\mathbf{S}_T(\mathbb{X})}$, there exists $(d', l', x') \in \text{norm}_{T, \mathbb{X}}(A)$ such that $(d', l', x') \vdash_{T, \mathbb{X}}^c (d, l, x)$. Now by definition of normalization, there exists $(d'', l'', x'') \in A$ such that $(d'', l'', x'') \equiv_{T, \mathbb{X}} (d', l', x')$. Then $(d'', l'', x'') \vdash_{T, \mathbb{X}} (d', l', x') \vdash_{T, \mathbb{X}}^c (d, l, x)$, and then $(d, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}(A)$. \square

Lemma E.3. $\forall \mathbb{X}, \mathbb{Y} \in |\mathbf{Alg}_\Gamma(\mathbf{C})|$ and $\forall h \in \mathbf{Alg}_\Gamma(\mathbf{C})[\mathbb{X}, \mathbb{Y}]$,

- (1) $\text{norm}_{T, \mathbb{X}}; d_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}} = d_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}$,
- (2) $\text{norm}_{T, \mathbb{X}}; \mathbf{H}(h); \text{norm}_{T, \mathbb{Y}} = \mathbf{H}(h); \text{norm}_{T, \mathbb{Y}}$,
- (3) $\text{norm}_{T, \mathbb{X}}$ is idempotent.

Proof. For the first point we prove that $\forall A \in |\mathbf{H}(\mathbb{X})|$ and $\forall c \in \Gamma$,

$$\begin{aligned} c_{\mathbf{S}_T(\mathbb{X}); \text{norm}_{T, \mathbb{X}}(A) &= \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A). \\ c_{\mathbf{S}_T(\mathbb{X}); \text{norm}_{T, \mathbb{X}}(A) &\subseteq \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A) \end{aligned}$$

Suppose that $(e', l', x') \in c_{\mathbf{S}_T(\mathbb{X}); \text{norm}_{T, \mathbb{X}}(A)$, then there exists $(e, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}(A)$ such that:

- (1) $(e, l, x) \equiv_{T, \mathbb{X}} (e', l', x')$,
- (2) it is not redundant in $c_{\mathbf{S}_T(\mathbb{X})}(A)$.

By definition of $c_{\mathbf{S}_T(\mathbb{X})}$, there exists $(d_0, l_0, x_0) \in A$ such that $(d_0, l_0, x_0) \vdash_{T, \mathbb{X}}^c (e, l, x)$.

Now, by Lemma 7.13, there exists $(d'_0, l'_0, x'_0) \in \text{norm}_{T, \mathbb{X}}(A)$ that either dominates (d_0, l_0, x_0) or $(d'_0, l'_0, x'_0) = (d_0, l_0, x_0)$. From definition of $c_{\mathbf{S}_T(\mathbb{X})}$, it follows that $(e, l, x) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$. Now we have directly that $(e, l, x) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$. Indeed, suppose ab absurdum that $(e, l, x) \notin \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$, then there exists a $(e_1, l_1, x_1) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$ that dominates (e, l, x) . Now, by Lemma E.2, we have also that $(e_1, l_1, x_1) \in c_{\mathbf{S}_T(\mathbb{X})}(A)$ that leads to absurd with 2.

Then $(e, l, x) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$, and also $(e', l', x') \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$, since the normalization function closes w.r.t. all equivalent transitions.

$$\text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A) \subseteq c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$$

Suppose that $(e', l', x') \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$, then there exists

$$(e, l, x) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A) \text{ such that:}$$

- (1) $(e, l, x) \equiv_{T, \mathbb{X}} (e', l', x')$,
- (2) it is not redundant in $\text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$.

Now, by Lemma E.2, $(e, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}(A)$. Now we have that $(e, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$. Indeed, suppose ab absurdum that $(e, l, x) \notin c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$, then there exists a $(e_1, l_1, x_1) \in c_{\mathbf{S}_T(\mathbb{X})}(A)$ that dominates (e, l, x) . Now, by definition of $c_{\mathbf{S}_T(\mathbb{X})}$, $(d''_0, l''_0, x''_0) \in A$ such that $(d''_0, l''_0, x''_0) \vdash_{T, \mathbb{X}}^c (e_1, l_1, x_1)$. Now, by Lemma 7.13, and by $(d''_0, l''_0, x''_0) \in A$, it follows that $(d'''_0, l'''_0, x'''_0) \in \text{norm}_{T, \mathbb{X}}(A)$ that either dominates (d''_0, l''_0, x''_0) or $(d'''_0, l'''_0, x'''_0) = (d''_0, l''_0, x''_0)$. By definition of $c_{\mathbf{S}_T(\mathbb{X})}$, $(e_1, l_1, x_1) \in \text{norm}_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$ and this together with 2 leads to an absurd.

Thus $(e, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A)$, and since $(e, l, x) \equiv (e', l', x')$,

$$(e', l', x') \in c_{\mathbf{S}_T(\mathbb{X})}; \text{norm}_{T, \mathbb{X}}(A).$$

For the second point we prove that $\forall A \in \mathbf{H}(X)$,

$$\text{norm}_{T,\mathbb{X}}; \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A) = \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A).$$

$$\text{norm}_{T,\mathbb{X}}; \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A) \subseteq \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$$

Suppose that $(d', l', y') \in \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$. Then there exists

$$(d, l, y) \in \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h)(A) \text{ such that}$$

$$(1) (d, l, y) \equiv_{T,\mathbb{Y}} (d', l', y'),$$

$$(2) \text{ it is not redundant in } \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h)(A).$$

Then $\exists x \in X$ such that $h(x) = y$ and $(d, l, x) \in \text{norm}_{T,\mathbb{X}}(A)$ and then $\exists (d'', l'', x'') \in A$ such that $(d, l, x) \equiv_{T,\mathbb{X}} (d'', l'', x'')$ and $(d'', l'', h(x'')) \in \mathbf{H}(h)(A)$.

Now suppose ab absurdum that $(d'', l'', y'') \notin \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$ where $y'' = h(x'')$. Then $\exists (d_0, l_0, y_0) \in \mathbf{H}(h)(A)$ such that $(d_0, l_0, y_0) \prec_{T,\mathbb{Y}} (d'', l'', y'')$. However, if $(d_0, l_0, y_0) \in \mathbf{H}(h)(A)$, then $(d_0, l_0, x_0) \in A$ such that $h(x_0) = y_0$ and by Lemma 7.13 there exists $(d'_0, l'_0, x'_0) \in \text{norm}_{T,\mathbb{X}}(A)$ that either dominates (d_0, l_0, x_0) or $(d'_0, l'_0, x'_0) = (d_0, l_0, x_0)$. By Lemma D.2, we have that $(d'_0, l'_0, h(x'_0)) \vdash_{T,\mathbb{Y}} (d_0, l_0, h(x_0)) \prec_{T,\mathbb{Y}} (d'', l'', y'')$ and, by Lemma E.1.1, $(d'_0, l'_0, h(x'_0)) \prec_{T,\mathbb{Y}} (d'', l'', y'') \equiv_{T,\mathbb{Y}} (d, l, y)$. Since $(d'_0, l'_0, h(x'_0)) \in \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h)(A)$, this leads to an absurdum.

Now we have $(d'', l'', y'') \in \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$ and $(d'', l'', y'') \equiv_{T,\mathbb{Y}} (d, l, y) \equiv_{T,\mathbb{Y}} (d', l', y')$ and, since $\text{norm}_{T,\mathbb{Y}}$ closes w.r.t. all equivalent transitions,

$$(d', l', y') \in \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A).$$

$$\mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A) \subseteq \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$$

Suppose that $(d', l', y') \in \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$, then there exists $(d, l, y) \in \mathbf{H}(h)(A)$, such that:

$$(1) (d, l, y) \equiv_{T,\mathbb{Y}} (d', l', y'),$$

$$(2) \text{ it is not redundant in } \mathbf{H}(h)(A).$$

Then $\exists x \in \mathbb{X}$, such that $h(x) = y$ and $(d, l, x) \in A$.

By Lemma 7.13, $\exists (d_0, l_0, x_0) \in \text{norm}_{T,\mathbb{X}}(A)$ (and $(d_0, l_0, x_0) \in A$) that either dominates (d, l, x) or $(d_0, l_0, x_0) = (d, l, x)$, and by Lemma D.2 $(d_0, l_0, h(x_0)) \vdash_{T,\mathbb{Y}} (d, l, h(x))$. Now we have two possible cases: or $(d, l, h(x)) \not\prec_{T,\mathbb{Y}} (d_0, l_0, h(x_0))$, or $(d, l, h(x)) \vdash_{T,\mathbb{Y}} (d_0, l_0, h(x_0))$. In the first case we have that $(d_0, l_0, h(x_0)) \prec_{T,\mathbb{Y}} (d, l, h(x))$, and this lead to absurdum with 2. Then, only the latter is possible, i.e., $(d_0, l_0, h(x_0)) \equiv_{T,\mathbb{Y}} (d, l, h(x))$.

Now suppose ab absurdum that $(d_0, l_0, h(x_0)) \notin \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$. Then $\exists (d_1, l_1, y_1) \in \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h)(A)$ that dominates $(d_0, l_0, h(x_0))$. Thus $\exists x_1 \in \mathbb{X}$ such that $h(x_1) = y_1$ and $(d_1, l_1, x_1) \in \text{norm}_{T,\mathbb{X}}(A)$ and $(d'_1, l'_1, x'_1) \in A$ such that $(d'_1, l'_1, x'_1) \equiv_{T,\mathbb{X}} (d_1, l_1, x_1)$.

Thus $(d'_1, l'_1, h(x'_1)) \in \mathbf{H}(h)(A)$ and

$$(d'_1, l'_1, h(x'_1)) \equiv_{T,\mathbb{Y}} (d_1, l_1, y_1) \prec_{T,\mathbb{Y}} (d_0, l_0, h(x_0)) \equiv_{T,\mathbb{Y}} (d, l, y),$$

i.e., $(d'_1, l'_1, h(x'_1)) \prec_{T,\mathbb{Y}} (d, l, y)$, against 2.

Then we have $(d_0, l_0, h(x_0)) \in \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$ and then also $(d', l', y') \in \text{norm}_{T,\mathbb{X}}; \mathbf{H}(h); \text{norm}_{T,\mathbb{Y}}(A)$.

For the third point we prove that $\forall A \in \mathbf{N}_T(X)$, $\text{norm}_{T,\mathbb{X}}(A) = A$. This is trivial, since $\text{norm}_{T,\mathbb{X}}$ junks away all the redundant transitions and add all those equivalent. But

since A is normalized, it does not contain any redundant transitions, and it is still closed by equivalent transitions. \square

Proposition E.4. $\mathbf{N}_T : \mathbf{Alg}_{\Gamma(\mathbf{C})} \rightarrow \mathbf{Alg}_{\Gamma(\mathbf{C})}$ is a functor.

Proof. First of all we have to prove that $\forall \mathbb{X} \in \mathbf{Alg}_{\Gamma(\mathbf{C})}$, $\mathbf{N}_T(\mathbb{X})$ is a $\Gamma(\mathbf{C})$ -algebra. In order to prove that, it is enough to show that $(c; d)_{\mathbf{N}_T(\mathbb{X})} = c_{\mathbf{N}_T(\mathbb{X})}; d_{\mathbf{N}_T(\mathbb{X})}$ and that $id_{\mathbf{N}_T(\mathbb{X})}$ is the identity function.

For the former, notice that $(c; d)_{\mathbf{N}_T(\mathbb{X})} = (c; d)_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}} = c_{\mathbf{S}_T(\mathbb{X})}; d_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}}$ since $\mathbf{S}_T(\mathbb{X})$ is a $\Gamma(\mathbf{C})$ -algebra. Now, By Lemma E.3.1, we have that it is equal to

$$c_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}}; d_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}},$$

i.e., $c_{\mathbf{N}_T(\mathbb{X})}; d_{\mathbf{N}_T(\mathbb{X})}$.

For the latter, notice that applying $id_{\mathbf{N}_T(\mathbb{X})}$ to a set of transitions A , it is the same of closing A w.r.t. the derivation relation $\vdash_{T, \mathbb{X}}$ and then normalizing it. Now, if A is normalized, one can close it w.r.t $\vdash_{T, \mathbb{X}}$, and then normalize it, obtaining the same set A . This is formally proved by Proposition 7.14.

Now we prove that $\mathbf{N}_T(h)$ is still a $\Gamma(\mathbf{C})$ -homomorphism. Recall that $\mathbf{H}(h) = \mathbf{S}_T(h)$,

$$c_{\mathbf{N}_T(\mathbb{X})}; \mathbf{N}_T(h) = c_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}}; \mathbf{N}_T(h) = c_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}}; \mathbf{S}_T(h); norm_{T, \mathbb{Y}} =$$

(by Lemma E.3.2)

$$c_{\mathbf{S}_T(\mathbb{X})}; \mathbf{S}_T(h); norm_{T, \mathbb{Y}} = \mathbf{S}_T(h); c_{\mathbf{S}_T(\mathbb{Y})}; norm_{T, \mathbb{Y}} =$$

(by Lemma E.3.1)

$$\mathbf{S}_T(h); norm_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})} norm_{T, \mathbb{Y}} = \mathbf{N}_T(h); c_{\mathbf{S}_T(\mathbb{Y})}; norm_{T, \mathbb{Y}} = \mathbf{N}_T(h); c_{\mathbf{N}_T(\mathbb{X})}.$$

In order to prove that $\mathbf{N}_T(id_{\mathbb{X}}) = id_{\mathbf{N}_T(\mathbb{X})}$ it is enough to observe that $\mathbf{N}_T(id_{\mathbb{X}}) = \mathbf{H}(id_{\mathbb{X}}); norm_{T, \mathbb{X}} = id_{\mathbf{H}(\mathbb{X})}; norm_{T, \mathbb{X}}$. Since in $\mathbf{N}_T(\mathbb{X})$ all the elements are normalized, by Lemma E.3.3, normalization plays no role.

In order to prove that \mathbf{N}_T preserves composition we use Lemma E.3.2: $\forall h : \mathbb{X} \rightarrow \mathbb{Y}, g : \mathbb{Y} \rightarrow \mathbb{Z}$,

$$\begin{aligned} \mathbf{N}_T(h; g) &= \mathbf{H}(h; g); norm_{T, \mathbb{Z}} = \mathbf{H}(h); \mathbf{H}(g); norm_{T, \mathbb{Z}} \\ &= \mathbf{H}(h); norm_{T, \mathbb{Y}}; \mathbf{H}(g); norm_{T, \mathbb{Z}} = \mathbf{N}_T(h); \mathbf{N}_T(g). \end{aligned} \quad \square$$

At the end of the appendix we prove the main theorem. Note that proof of Lemma 7.13 is in Appendix E.

Lemma E.5. $norm_{T, \mathbb{X}} : \mathbf{S}_T(\mathbb{X}) \rightarrow \mathbf{N}_T(\mathbb{X})$ and $sat_{T, \mathbb{X}} : \mathbf{N}_T(\mathbb{X}) \rightarrow \mathbf{S}_T(\mathbb{X})$ are $\Gamma(\mathbf{C})$ -homomorphisms.

Proof. For all operators c , we have that

$$c_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}} = \text{by Lemma E.3.1} = norm_{T, \mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}} = norm_{T, \mathbb{X}}; c_{\mathbf{N}_T(\mathbb{X})}.$$

For $sat_{T, \mathbb{X}}$ we have that $c_{\mathbf{N}_T(\mathbb{X})}; sat_{T, \mathbb{X}} = c_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}}; sat_{T, \mathbb{X}}$. Note that

$$c_{\mathbf{S}_T(\mathbb{X})}; norm_{T, \mathbb{X}}; sat_{T, \mathbb{X}} = c_{\mathbf{S}_T(\mathbb{X})}; sat_{T, \mathbb{X}}$$

since saturation adds everything that is removed by normalization. At this point, it is enough to prove that $c_{\mathbf{S}_T(\mathbb{X})}; sat_{T,\mathbb{X}} = sat_{T,\mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}$.

We have to prove that $\forall A \in |\mathbf{H}(\mathbb{X})|, c_{\mathbf{S}_T(\mathbb{X})}; sat_{T,\mathbb{X}}(A) = sat_{T,\mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$.

$$c_{\mathbf{S}_T(\mathbb{X})}; sat_{T,\mathbb{X}}(A) \subseteq sat_{T,\mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$$

Suppose that $(e, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}; sat_{T,\mathbb{X}}(A)$, then there exists $(e', l', x') \in c_{\mathbf{S}_T(\mathbb{X})}(A)$ such that $(e', l', x') \vdash_{T,\mathbb{X}} (e, l, x)$, and by definition of $c_{\mathbf{S}_T(\mathbb{X})}$, there exists $(e'_0, l'_0, x'_0) \in A$ (and then also in $sat_{T,\mathbb{X}}(A)$) such that $(e'_0, l'_0, x'_0) \vdash_{T,\mathbb{X}}^c (e', l', x') \vdash_{T,\mathbb{X}} (e, l, x)$. Then $(e'_0, l'_0, x'_0) \vdash_{T,\mathbb{X}}^c (e, l, x)$, and then $(e, l, x) \in sat_{T,\mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$.

$$sat_{T,\mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A) \subseteq c_{\mathbf{S}_T(\mathbb{X})}; sat_{T,\mathbb{X}}(A)$$

Suppose that $(e, l, x) \in sat_{T,\mathbb{X}}; c_{\mathbf{S}_T(\mathbb{X})}(A)$, then there exists $(d', l', x') \in sat_{T,\mathbb{X}}(A)$ such that $(d', l', x') \vdash_{T,\mathbb{X}}^c (e, l, x)$. Thus, by definition of $sat_{T,\mathbb{X}}$, there exists $(d'', l'', x'') \in A$ such that $(d'', l'', x'') \vdash_{T,\mathbb{X}} (d', l', x')$. Then $(d'', l'', x'') \vdash_{T,\mathbb{X}}^c (e, l, x)$ and then $(e, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}; sat_{T,\mathbb{X}}(A)$, and then $(e, l, x) \in c_{\mathbf{S}_T(\mathbb{X})}; sat_{T,\mathbb{X}}(A)$. \square

Lemma E.6. *norm_T and sat_T are isomorphisms, one the inverse of the other.*

Proof. Since by Lemma E.5, $norm_{T,\mathbb{X}}$ and $sat_{T,\mathbb{X}}$ are morphisms in $\mathbf{Alg}_\Gamma(\mathbf{C})$, we have just to prove that $norm_{T,\mathbb{X}}; sat_{T,\mathbb{X}} = id_{\mathbf{S}_T(\mathbb{X})}$ and $sat_{T,\mathbb{X}}; norm_{T,\mathbb{X}} = id_{\mathbf{N}_T(\mathbb{X})}$.

$$norm_{T,\mathbb{X}}; sat_{T,\mathbb{X}}(A) \subseteq A$$

If $(d, l, x) \in norm_{T,\mathbb{X}}; sat_{T,\mathbb{X}}(A)$, then $(d', l', x') \in norm_{T,\mathbb{X}}(A)$ such that $(d', l', x') \vdash_{T,\mathbb{X}} (d, l, x)$. Thus $(d'', l'', x'') \in A$ such that $(d'', l'', x'') \equiv_{T,\mathbb{X}} (d', l', x')$. Then $(d'', l'', x'') \vdash_{T,\mathbb{X}} (d, l, x)$. Now, also $(d, l, x) \in A$, since A is saturated.

$$A \subseteq norm_{T,\mathbb{X}}; sat_{T,\mathbb{X}}(A)$$

If $(d, l, x) \in A$ then, by Lemma 7.13, there exists $(d', l', x') \in norm_{T,\mathbb{X}}(A)$ that either dominates (d, l, x) or $(d', l', x') = (d, l, x)$. Thus $(d, l, x) \in norm_{T,\mathbb{X}}; sat_{T,\mathbb{X}}(A)$.

$$sat_{T,\mathbb{X}}; norm_{T,\mathbb{X}}(A) \subseteq A$$

If $(d', l', x') \in sat_{T,\mathbb{X}}; norm_{T,\mathbb{X}}(A)$ then there exist $(d, l, x) \in sat_{T,\mathbb{X}}(A)$ such that

- (1) $(d, l, x) \equiv_{T,\mathbb{X}} (d', l', x')$,
- (2) it is not redundant in $sat_{T,\mathbb{X}}(A)$.

Then $\exists (d_0, l_0, x_0) \in A$ such that $(d_0, l_0, x_0) \vdash_{T,\mathbb{X}} (d, l, x)$. Now we have two possibilities. Firstly, $(d, l, x) \not\vdash_{T,\mathbb{X}} (d_0, l_0, x_0)$, then $(d_0, l_0, x_0) \prec_{T,\mathbb{X}} (d, l, x)$ and this is absurd with 2. Secondly, $(d, l, x) \vdash_{T,\mathbb{X}} (d_0, l_0, x_0)$ and then $(d, l, x) \equiv_{T,\mathbb{X}} (d_0, l_0, x_0)$, and since A is normalized, $(d, l, x) \in A$.

$$A \subseteq sat_{T,\mathbb{X}}; norm_{T,\mathbb{X}}(A)$$

If $(d, l, x) \in A$, then $(d, l, x) \in sat_{T,\mathbb{X}}(A)$. Now suppose ab absurdum that $(d, l, x) \notin sat_{T,\mathbb{X}}; norm_{T,\mathbb{X}}(A)$ then there exists a $(d', l', x') \in sat_{T,\mathbb{X}}(A)$ that dominates (d, l, x) . Then, by definition of $sat_{T,\mathbb{X}}$, $(d'', l'', x'') \in A$ that dominates (d', l', x') . But then (d'', l'', x'') dominates also (d, l, x) , against the hypothesis that A is normalized. \square

$$\begin{array}{ccccc}
\mathbb{X} & \xrightarrow{h} & \mathbb{Y} & & \mathbf{S}_{\mathbf{T}}(\mathbb{X}) & \xrightarrow{\mathbf{S}_{\mathbf{T}}(h)} & \mathbf{S}_{\mathbf{T}}(\mathbb{Y}) & & \mathbf{N}_{\mathbf{T}}(\mathbb{X}) & \xrightarrow{\mathbf{N}_{\mathbf{T}}(h)} & \mathbf{N}_{\mathbf{T}}(\mathbb{Y}) \\
\downarrow \alpha & & \downarrow \beta & & \downarrow \text{norm}_{T,\mathbb{X}} & & \downarrow \text{norm}_{T,\mathbb{Y}} & & \downarrow \text{sat}_{T,\mathbb{X}} & & \downarrow \text{sat}_{T,\mathbb{Y}} \\
\mathbf{S}_{\mathbf{T}}(\mathbb{X}) & \xrightarrow{\mathbf{S}_{\mathbf{T}}(h)} & \mathbf{S}_{\mathbf{T}}(\mathbb{Y}) & & \mathbf{N}_{\mathbf{T}}(\mathbb{X}) & \xrightarrow{\mathbf{N}_{\mathbf{T}}(h)} & \mathbf{N}_{\mathbf{T}}(\mathbb{Y}) & & \mathbf{S}_{\mathbf{T}}(\mathbb{X}) & \xrightarrow{\mathbf{S}_{\mathbf{T}}(h)} & \mathbf{S}_{\mathbf{T}}(\mathbb{Y}) \\
& & \text{(i)} & & & & \text{(ii)} & & & & \text{(iii)}
\end{array}$$

Figure 13: norm_T and sat_T are natural transformations.

Proposition 7.14. *Let norm_T , respectively, sat_T be the families of morphisms $\{\text{norm}_{T,\mathbb{X}} : \mathbf{S}_{\mathbf{T}}(\mathbb{X}) \rightarrow \mathbf{N}_{\mathbf{T}}(\mathbb{X}), \forall \mathbb{X} \in |\mathbf{Alg}_{\Gamma(\mathbf{C})}|\}$ and $\{\text{sat}_{T,\mathbb{X}} : \mathbf{N}_{\mathbf{T}}(\mathbb{X}) \rightarrow \mathbf{S}_{\mathbf{T}}(\mathbb{X}), \forall \mathbb{X} \in |\mathbf{Alg}_{\Gamma(\mathbf{C})}|\}$. Then $\text{norm}_T : \mathbf{S}_{\mathbf{T}} \Rightarrow \mathbf{N}_{\mathbf{T}}$ and $\text{sat}_T : \mathbf{N}_{\mathbf{T}} \Rightarrow \mathbf{S}_{\mathbf{T}}$ are natural transformations. More precisely, they are natural isomorphisms, one the inverse of the other.*

Proof. Since by Lemma E.6, $\text{norm}_{T,\mathbb{X}}$ and $\text{sat}_{T,\mathbb{X}}$ are one the inverse of the other, we have only to prove that they are natural transformation, i.e., that diagrams (ii) and (iii) in Figure 13 commute. Notes that by definition, $\mathbf{N}_{\mathbf{T}}(h) = \mathbf{H}(h)$; $\text{norm}_{T,\mathbb{Y}}$ and thus diagram (ii) commutes by Lemma E.3.2.

Then, by Lemma E.6, also diagram (iii) commutes. \square

Theorem 7.15. *$\mathbf{Coalg}_{\mathbf{S}_{\mathbf{T}}}$ and $\mathbf{Coalg}_{\mathbf{N}_{\mathbf{T}}}$ are isomorphic.*

Proof. Let $\mathbf{NORM}_{\mathbf{T}} : \mathbf{Coalg}_{\mathbf{S}_{\mathbf{T}}} \rightarrow \mathbf{Coalg}_{\mathbf{N}_{\mathbf{T}}}$ be the functor sending an object $\langle \mathbb{X}, \alpha \rangle$ into $\langle \mathbb{X}, \alpha; \text{norm}_{T,\mathbb{X}} \rangle$ and any morphism h to itself. Let $\mathbf{SAT}_{\mathbf{T}} : \mathbf{Coalg}_{\mathbf{N}_{\mathbf{T}}} \rightarrow \mathbf{Coalg}_{\mathbf{S}_{\mathbf{T}}}$ be the functor sending $\langle \mathbb{X}, \alpha \rangle$ into $\langle \mathbb{X}, \alpha; \text{sat}_{T,\mathbb{X}} \rangle$ and any morphism h in itself. By Proposition 7.14, these are clearly, one the inverse of the other. \square

APPENDIX F. FACTORIZATION SYSTEM FOR \mathbf{H} -COALGEBRAS

The notions of *subcoalgebra* and *homomorphic image* have been introduced in [40], for coalgebras over \mathbf{Set} . These notions have been extended by Kurz in his thesis [28] to coalgebras over a generic category \mathbf{C} , by employing factorization systems.

As subcolagebras and homomorphic images are fundamental for proving that $|\mathbf{Coalg}_{\mathbf{S}_{\mathbf{T}}}|$ is a covariety of $\mathbf{Coalg}_{\mathbf{H}}$ (and thus proving the existence of final system), we briefly report here these definitions.

Definition F.1 (Factorization System). Let \mathbf{C} be some category, and let E, M be classes of morphisms in \mathbf{C} . Then (E, M) is a factorization system for \mathbf{C} if and only if

- (1) E, M are closed under isomorphism,
- (2) \mathbf{C} has (E, M) -factorizations, i.e., every morphism f in \mathbf{C} has a factorization $f = e; m$ for $e \in E$ and $m \in M$,
- (3) \mathbf{C} has the unique (E, M) -diagonalisation property, i.e., whenever the square

$$\begin{array}{ccc}
A & \xrightarrow{e} & B \\
\downarrow f & \nearrow d & \downarrow g \\
C & \xrightarrow{m} & D
\end{array}$$

commutes for $m \in M$ and $e \in E$, then there is a unique diagonal d making the two triangle commute.

The theory of coalgebras has been mainly developed for coalgebras over **Set**. In Section 1.4 of [28], Kurz generalizes this theory for coalgebras over a generic category **C**, by providing four axioms relying on a factorization system for **C** and some properties of the endofunctor. These axioms guarantees that the resulting category has all the good qualities of coalgebras over **Set**, such as, for example, that the collection of all subcoalgebras of a coalgebra is a complete lattice.

It can be easily proved (looking at $\mathbf{Alg}_{\Gamma(\mathbf{C})}$ as $\mathbf{Set}^{\mathbf{C}}$) that the endofunctor **H** satisfies these four axioms when considering the following factorization system.

Definition F.2. The factorization system for $\mathbf{Alg}_{\Gamma(\mathbf{C})}$ is $(E_{\mathbf{C}}, M_{\mathbf{C}})$, where $E_{\mathbf{C}}$ is the class of $|\mathbf{C}|$ -indexed homomorphism having all components epi, while $M_{\mathbf{C}}$ is the class of $|\mathbf{C}|$ -indexed homomorphism having all components mono.

Here, we want to show that the forgetful functor $\mathbf{U} : \mathbf{Coalgs}_{\mathbf{T}} \rightarrow \mathbf{Alg}_{\Gamma(\mathbf{C})}$ creates factorizations with respect to $(E_{\mathbf{C}}, M_{\mathbf{C}})$ (Axiom 1.2). This means that if $h : \langle \mathbb{X}, \alpha \rangle \rightarrow \langle \mathbb{Y}, \beta \rangle$ is a morphism in $\mathbf{Coalgs}_{\mathbf{T}}$ and $h = e; m$ is a factorization in $(E_{\mathbf{C}}, M_{\mathbf{C}})$, then it is also a factorization in $\mathbf{Coalgs}_{\mathbf{T}}$, i.e., e, m are also cohomomorphisms. This is graphically depicted below.

$$\begin{array}{ccc}
 \mathbb{X} & \xrightarrow{h} & \mathbb{Y} \\
 \searrow e & & \nearrow m \\
 & \mathbb{I} & \\
 \alpha \downarrow & \text{---} \gamma \text{---} & \downarrow \beta \\
 \mathbf{H}(\mathbb{X}) & \xrightarrow{\mathbf{H}(h)} & \mathbf{H}(\mathbb{Y}) \\
 \searrow \mathbf{H}(e) & & \nearrow \mathbf{H}(m) \\
 & \mathbf{H}(\mathbb{I}) &
 \end{array}$$

If the back square commutes and $h = e; m$ is factorization with respect to $(E_{\mathbf{C}}, M_{\mathbf{C}})$, then also $\mathbf{H}(e)$ is in $E_{\mathbf{C}}$ and $\mathbf{H}(m)$ is in $M_{\mathbf{C}}$. The unique arrow γ comes from the diagonalization property noting that:

$$\begin{array}{ccc}
 \mathbb{X} & \xrightarrow{e} & \mathbb{I} \\
 \alpha; \mathbf{H}(e) \downarrow & \nearrow \gamma & \downarrow m; \beta \\
 \mathbf{H}(\mathbb{I}) & \xrightarrow{\mathbf{H}(m)} & \mathbf{H}(\mathbb{Y})
 \end{array}$$

At this point we can define subcoalgebra and homomorphic image.

Definition F.3 (Subcoalgebra). Let $m : \langle \mathbb{X}, \alpha \rangle \rightarrow \langle \mathbb{Y}, \beta \rangle$ be an arrow of $\mathbf{Coalg}_{\mathbf{H}}$. Then $\langle \mathbb{X}, \alpha \rangle$ is said a subcoalgebra of $\langle \mathbb{Y}, \beta \rangle$ if $m \in M_{\mathbf{C}}$.

Definition F.4 (Homomorphic Image). Let $f : \langle \mathbb{X}, \alpha \rangle \rightarrow \langle \mathbb{Y}, \beta \rangle$ be an arrow of $\mathbf{Coalg}_{\mathbf{H}}$. The homomorphic image of $\langle \mathbb{X}, \alpha \rangle$ through f is the coalgebra $\langle \mathbb{I}, \gamma \rangle$ shown in the diagram above.