Coalgebraic Models for Reactive Systems*

Filippo Bonchi and Ugo Montanari

Dipartimento di Informatica, Università di Pisa

Abstract. Reactive Systems *à la* Leifer and Milner allow to derive from a reaction semantics definition an LTS equipped with a bisimilarity relation which is a congruence. This theory has been extended by the authors (together with Barbara König) in order to handle saturated bisimilarity, a coarser equivalence that is more adequate for some interesting formalisms, such as logic programming and open pi-calculus. In this paper we recast the theory of Reactive Systems inside Universal Coalgebra. This construction is particularly useful for saturated bisimilarity, which can be seen as final semantics of Normalized Coalgebras. These are structured coalgebras (not bialgebras) where the sets of transitions are minimized rather than maximized as in saturated LTS, still yielding the same semantics. We give evidence the effectiveness of our approach minimizing an Open Petri net in a category of Normalized Coalgebras.

1 Introduction

The operational semantics of process calculi has traditionally been specified by labelled transition systems (LTSs), and the abstract semantics by bisimilarity relations defined on them. Bisimilarities often turn out to be congruences with respect to the operations of the languages, a property which expresses the compositionality of the abstract semantics. A simpler approach, inspired by classical formalisms like λ-calculus, Petri nets, term and graph rewriting - pioneered by the Chemical Abstract Machine [3] and especially convenient for nominal calculi - defines operational semantics by means of *structural axioms* and *reaction rules*. Transitions caused by reaction rules, however, are not labeled, since they represent evolutions of the system without interactions with the external world. Thus reaction semantics is neither abstract nor compositional.

To enhance the expressiveness of reaction semantics, Leifer and Milner proposed in [12] the *theory of reactive systems*: a systematic method for deriving a labeled transition system from reaction rules. The main idea is the following: a process p can do a move with label $C[-]$ and become p' iff there is a reaction rule transforming $C[p]$ in p'. This LTS is called *Context Transitions System* (*CTS*) and the bisimilarity over it ($\sim_{SAT}$, called *saturated*) is always a congruence. However, such an LTS is usually very large, typically infinite branching and overloaded with redundant transitions, since often contexts $C[-]$ contain components which are irrelevant for the transition.

For this reason, Leifer and Milner introduced the notions of relative pushout (RPO) and idem relative pushout (IPO) for specifying a/the minimal context that allows the

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

L. Caires and V.T. Vasconcelos (Eds.): CONCUR 2007, LNCS 4703, pp. 364–379, 2007.

state to react with a rule. This construction leads to the *IPO transition system* (*ITS*), that uses only contexts generated by IPOs, and not all contexts, as labels, and thus is smaller than *CTS*. Bisimilarity on this LTS ($\sim_{IPO}$) is a congruence under restrictive conditions.

In [4], the authors proved that for some interesting formalisms, such as logic programming and open π-calculus, $\sim_{IPO}$ is at some extent inadequate, since it is strictly finer than standard abstract semantics, while $\sim_{SAT}$ exactly characterizes it.

Universal Coalgebra [15] provides a categorical framework where abstract semantics of interactive computing systems are described as morphisms to their minimal representatives. More precisely, given an endofunctor $\mathbf{F}$ on a category $\mathbf{C}$, a coalgebra is an arrow $f\colon X \to \mathbf{F}(X)$ of $\mathbf{C}$ and a coalgebra morphism from f to f' is an arrow $h\colon X \to X'$ of $\mathbf{C}$ with $h\,;f' = f\,;\mathbf{F}(h)$. Under certain conditions on $\mathbf{C}$ and $\mathbf{F}$, a category of coalgebras admits a final object. Ordinary labeled transition systems (with finite or countable branching) can be represented as coalgebras with final object for a suitable functor on $\mathbf{Set}$. Then, in order to prove that two states are equivalent, we have to check if they are identified by the final morphism, and the image of the given coalgebra through the latter is the minimal representative, which in the finite case can be usually computed via the list partitioning algorithm by Kanellakis and Smolka [9].

However, this representation of interactive systems forgets about the algebraic structure, which is usually very relevant in practical cases, since compositionality is the key to master complexity. In particular, the property that bisimilarity respects the operations, i.e. that it is a congruence, which is essential for making abstract semantics compositional, is not reflected in the structure of the model.

In [19], *bialgebras* are introduced as a model with both algebraic and coalgebraic structure, while a related approach based on *structured coalgebras* is presented in [7]. In the latter work, the endofunctor determining the coalgebraic structure is lifted from $\mathbf{Set}$ to the category of Σ-algebras, for some algebraic signature Σ. Morphisms between coalgebras in this category are both Σ-homomorphisms and coalgebra morphisms: as a consequence the unique morphism to the final coalgebra always induces a bisimilarity that is a congruence.

In this paper we provide a structured coalgebraic construction for both *ITS* and *CTS*. This is interesting for at least two reasons. On the one hand it assures the existence of final semantics and minimal representatives for reactive systems, both for the Leifer and Milner's IPO semantics $\sim_{IPO}$ and for our saturated semantics $\sim_{SAT}$. On the other hand it is an alternative compositionality proof of them.

For practical applications a key issue is how efficiently our saturated semantics can be computed, which, according to its definition, is based on the large and redundant *CTS*. In the previous paper [4], an unconventional notion of bisimulation, called *semisaturated*, is presented for this purpose. It allows Alice, the first player of the bisimulation game, to choose a transition in *ITS*, while Bob, the second player, chooses in *CTS*. Semisaturated bisimilarity is the same as saturated bisimilarity, but the size of the game is much smaller. Unfortunately, this approach cannot be extended to coalgebraic theory, since in the latter case there is only one transition system for both players.

A further contribution of the paper is the construction of yet another LTS which has fewer transitions than *ITS*, but supports $\sim_{SAT}$. In reactive systems all non-IPO transitions, i.e. the transitions labeled with a context that is not strictly necessary to perform a transition with a given rule, are considered redundant and omitted. Here we introduce a stronger notion of redundancy. Indeed we consider redundant the transitions $p \overset{C[-]}{\to} p'$ such that $p \overset{D[-]}{\to} p''$, and $D[-], p''$ are smaller, i.e., there exists a context $E[-]$ such that $E[D[-]] = C[-]$ and $E[p''] \sim p'$. Thus our notion of redundancy is based on bisimilarity and it is independent from the rule that allows the reaction, while the IPO construction is based on syntactic equivalence and it is relative to a particular rule.

Our construction is based on *Normalized Coalgebras*. These are structured coalgebras without redundant transitions which form a category with final object, where the unique morphism induces a notion of bisimilarity completely abstract from redundant transitions. We prove that the category of Normalized Coalgebras is isomorphic to the category of saturated coalgebras (the coalgebras containing all the redundant transitions), where the large context transition system *CTS* can be directly modelled. In doing this, we use the notions of *normalization* that junks away all the redundant transitions, and of *saturation* that adds all the redundant transitions. Both are natural transformations between the functors (defining the two categories of coalgebras) and one is the inverse of the other. As a corollary of the isomorphism theorem, $\sim_{SAT}$ can be characterized as bisimilarity in the category of Normalized Coalgebras. This proves that our notion of non-redundancy is more canonical than IPOs, since it exactly captures $\sim_{SAT}$.

Normalized Coalgebras provide an efficient way to compute $\sim_{SAT}$. Indeed we can forget about all the redundant transitions (obtaining a labeled transition system smaller than *ITS*) and then we can compute the final morphism in the category of normalized coalgebras, through the canonical minimization algorithm. Normalized Coalgebras turn out to be theoretically interesting for one additional reason. Those are, to our knowledge, the first interesting example of structured coalgebras that are not bialgebras.

Synopsis. In Sec. 2 and 3, we introduce the theory of reactive systems and (structured) coalgebras. Then in Sec. 4 and 5 we provide a structured coalgebraic construction for *CTS* and for *ITS*. In Sec. 6 we introduce Normalized Coalgebras, a minimization algorithm for these and we apply it to a concrete example.

2 The Theory of Reactive Systems

Here we summarize the theory of reactive systems proposed in [12] to derive labelled transition systems and bisimulation congruences from a given reaction semantics. The theory is centred on the concepts of *term*, *context* and *reaction rules*: contexts are arrows of a category, terms are arrows having as domain 0 (a special object that denotes no holes), and reaction rules are pairs of terms.

Definition 1 (Reactive System). *A* reactive system $\mathcal{R}$ *consists of:*

1. *a category* $\mathbf{C}$
2. *a distinguished object* $0 \in |\mathbf{C}|$
3. *a composition-reflecting subcategory* $\mathbf{D}$ *of* reactive contexts
4. *a set of pairs* $\mathbb{R} \subseteq \bigcup_{m \in |\mathbf{C}|} \mathbf{C}[0, m] \times \mathbf{C}[0, m]$ *of* reaction rules.

The reactive contexts are those in which a reaction can occur. By composition-reflecting we mean that $d; d' \in \mathbf{D}$ implies $d, d' \in \mathbf{D}$.

From reaction rules one generates the reaction relation by closing them under all reactive contexts. Formally, the *reaction relation* is defined by taking $p \rightsquigarrow q$ if there is $\langle l, r \rangle \in \mathbb{R}$ and $d \in \mathbf{D}$ such that $p = l; d$ and $q = r; d$.

Thus the behaviour of a reactive system is expressed as an unlabelled transition system. On the other hand many behavioural equivalences are only defined for LTSs. In order to obtain an LTS, we can plug a term p into some context c and observe if a reaction occurs. In this case we have that $p \stackrel{c}{\rightarrow}$. Categorically speaking this means that $p; c$ matches $l; d$ for some rule $\langle l, r \rangle \in \mathbb{R}$ and some reactive context d. This situation is depicted by diagram (i) in Fig. 1: a commuting diagram like this is said a *redex square*.

Definition 2. *The* context transition system *(CTS for short) is defined as follows:*

- *states: arrows* $p : 0 \rightarrow m$ *in* $\mathbf{C}$*, for arbitrary* m*;*
- *transitions:* $p \stackrel{c}{\rightarrow}_C q$ *iff* $p; c \rightsquigarrow q$*.*

Bisimilarity on this LTS is called *saturated* (denoted by $\sim_{SAT}$), and it is always a congruence (i.e., preserved under all contexts). However this labelled transition system is often infinite-branching since all contexts that allow reactions may occur as labels. Another problem of *CTS* is that it has redundant transitions. For example, consider the term $a.0$ of CCS. The observer can put this term into the context $\overline{a}.0 \mid -$ and observe a reaction. This corresponds to the transition $a.0 \stackrel{\overline{a}.0|-}{\rightarrow}_C 0|0$. However we also have $a.0 \stackrel{p|\overline{a}.0|-}{\rightarrow}_C p \mid 0 \mid 0$ as a transition, yet p does not contribute to the reaction. Hence we need a notion of "minimal context that allows a reaction". Leifer and Milner define idem pushouts (IPOs) in order to capture this notion.

Definition 3 (RPO/IPO). *Let the diagrams (ii)-(v) in Fig. 1 be in some category* $\mathbf{C}$*. Let (ii) be commuting. Any tuple* $\langle x, e, f, g \rangle$ *which makes (iii) commute is called a* candidate *for (ii). A* relative pushout (RPO) *is the smallest such candidate. More formally, it satisfies the universal property that given any other candidate* $\langle y, e', f', g' \rangle$*, there exists a unique mediating morphism* $h : x \rightarrow y$ *such that (iv) and (v) commute.*

Diagram (ii) of Fig. 1 is called idem pushout (IPO) *if* $\langle o, c, d, id_o \rangle$ *is an RPO.*

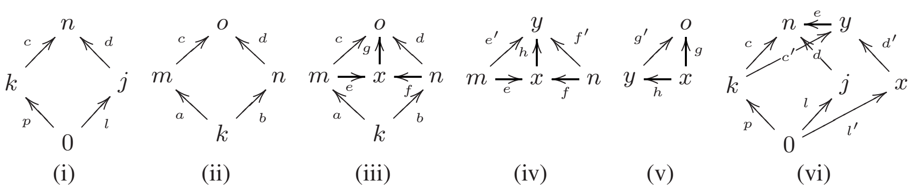


Fig. 1. Redex Square and RPO

We say that a reactive system *has (redex) RPOs* if, in the underlying category, for each (redex) square there exists an RPO, while it *has (redex) IPOs*, if every (redex) square has at least one IPO as candidate. A deeper discussion about the relationship between the two concepts can be found in [4].

Definition 4. *The* IPO transition system *(ITS for short) is defined as follows:*

- *states:* $p : 0 \to m$ *in* $\mathbf{C}$*, for arbitrary* m*;*
- *transitions:* $p \xrightarrow{c}_I r; d$ *iff* $d \in \mathbf{D}$, $\langle l, r \rangle \in \mathbb{R}$ *and the diagram (i) in Fig. 1 is an IPO.*

In other words, if inserting p into the context c matches $l; d$, and c is the "smallest" such context (according to the IPO condition), then p transforms to $r; d$ with label c, where r is the reduct of l. Bisimilarity on *ITS* is referred to as *standard bisimilarity* (denoted by $\sim_{IPO}$), and [12] proves the following.

Proposition 1. *In reactive systems with redex RPOs,* $\sim_{IPO}$ *is a congruence.*

In [4], the authors, together with Barbara König, show that $\sim_{IPO}$ is usually finer than $\sim_{SAT}$, and the latter is more appropriate than the former in some important cases: in Logic Programming and Open π-calculus, saturated semantics capture the canonical abstract semantics (i.e., logic equivalence and open bisimilarity), while standard semantics result too fine. Since *CTS* is full of redundancy (and usually infinite branching), the authors introduce semi saturated bisimulation to efficiently characterize $\sim_{SAT}$.

Definition 5 (Semi-Saturated Bisimulation). *A symmetric relation* R *is a semi-saturated bisimulation iff whenever* $p \, R \, q$*,*
if $p \xrightarrow{c}_I p'$ *then* $q \xrightarrow{d}_I q'$ *and* $\exists e \in \mathbf{D}$ *such that* $d; e = c$ *and* $p' \, R \, q'; e$*.*
The union of all Semi-Saturated bisimulations is Semi-Saturated bisimilarity *(*$\sim_{SS}$*).*

This characterization is more efficient than considering all the possible contexts as labels. Nevertheless, as the following proposition states, it coincides with $\sim_{SAT}$.

Proposition 2. *In reactive systems with redex IPOs,* $\sim_{SAT} = \sim_{SS}$ *.*

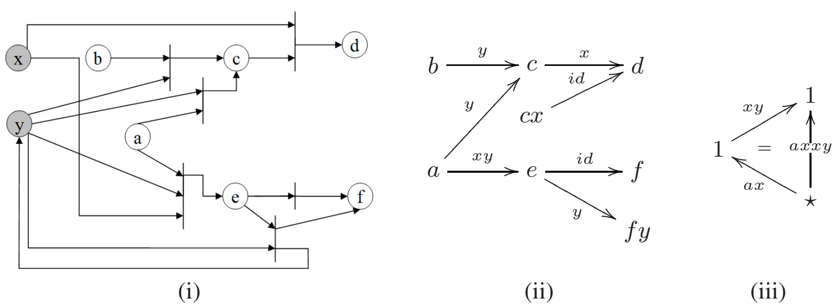


Fig. 2. (i) The Open Petri net $\mathcal{N}$. (ii) The *ITS* of a, b and cx. (iii) Arrows composition in $\mathbf{OPL}$.

Example 1. Open Petri nets [10,2] are P/T nets equipped with an *interface*, i.e., a set of *open places*, where nets can receive tokens from the environment[1]. Consider the Open net in Fig. 2(i). The interface of this set is the set of open places x and y depicted in gray. This net defines the reactive system $\mathcal{N} = \langle \mathbf{OPL}, \star, \mathbf{OPL}, \mathbb{T} \rangle$. Roughly the states of

[1] [13] encodes C/E nets into Bigraphs [14], while [16] P/T nets into Borrowed Contexts [8].

OPL (arrows from $\star$ to 1) are multisets on all the places, while contexts (arrows from 1 to 1) are multisets on open places. The composition of a state m_1 with a context m_2 is defined as the union of the multisets m_1 and m_2 as shown in Fig. 2(iii). Every transition of the net describes a reaction rule, where the left hand side is the precondition of the transition, and the right hand side is the postcondition.

Hereafterwe will use id for the empty multiset and aab for the multiset $\{a, a, b\}$. The *ITS* of a and b is depicted in Fig.2(ii). Consider the multisets e and cx. The former can interact both with the rule $\langle e, f\rangle$ generating the transition $e \xrightarrow{id}_I f$ and with the rule $\langle ey, fy\rangle$ generating the transition $e \xrightarrow{y}_I fy$. The latter can interact only with the rule $\langle cx, d\rangle$ generating the transition $cx \xrightarrow{id}_I d$. Thus $e \nsim_{IPO} cx$, but $e \sim_{SAT} cx$. Indeed the *CTS* move $e \xrightarrow{y}_C fy$ is matched by $cx \xrightarrow{y}_C dy$ and $fy \sim_{SAT} dy$ since both cannot move. Moreover $a \nsim_{IPO} b$ but $a \sim_{SS} b$ (and thus $a \sim_{SAT} b$). Indeed when a proposes $a \xrightarrow{xy}_I e$, b can answer with $b \xrightarrow{y}_I c$ and, as proved above, $e \sim_{SAT} cx$.

3 Coalgebras and Structured Coalgebras

In this section we introduce first the basic notions of the theory of coalgebras [15] and then structured coalgebras [7] in order to model reactive systems.

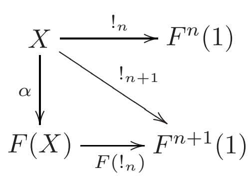
Definition 6 (coalgebras and cohomomorphisms). *Let* $\mathbf{F} : \mathbf{C} \to \mathbf{C}$ *be an endofunctor on a category* $\mathbf{C}$. *A* coalgebra for $\mathbf{F}$ *or (*$\mathbf{F}$-coalgebra*) is a pair* $\langle A, \alpha\rangle$ *where* A *is an object of* $\mathbf{C}$ *and* $\alpha : A \to \mathbf{F}(A)$ *is an arrow. An* $\mathbf{F}$-cohomomorphism $f : \langle A, \alpha\rangle \to \langle B, \beta\rangle$ *is an arrow* $f : A \to B$ *of* $\mathbf{C}$ *such that* $f; \beta = \alpha; \mathbf{F}(f)$.

$\mathbf{Coalg_F}$ *is the category of* $\mathbf{F}$*-coalgebras and* $\mathbf{F}$*-cohomomorphisms.*

Let $\mathbf{P_L} : \mathbf{Set} \to \mathbf{Set}$ be the functor defined as $X \mapsto \mathbf{P}(L \times X)$ where L is a fixed set of labels and $\mathbf{P}$ denotes the powerset functor. Then coalgebras for this functor are one-to-one with labeled transition systems over L [15]. Transition system morphisms are usually defined as functions between the carriers that *preserve* transitions, while $\mathbf{P_L}$-cohomomorphisms not only preserve, but also *reflect* transitions.

We can give a categorical characterization of bisimilarity if there exists a *final coalgebra*. Two elements of the carrier of a coalgebra are bisimilar iff they are mapped to the same element of the final coalgebra by the unique cohomomorphism. Indeed, in the final coalgebra all bisimilar states are identified, and thus, the image of a coalgebra through the unique morphism, is the minimal realization (w.r.t. bisimilarity) of the coalgebra. Computing the unique morphism just means to minimize the coalgebras, that it is usually possible in the finite case, using the following algorithm [1]:

1. Given a $\mathbf{F}$-coalgebra $\langle X, \alpha\rangle$, we initialize $!_0 : X \to 1$ as the morphism that maps all the elements of X into the one element set 1. This represents the trivial partitioning where all the elements are considered equivalent.
2. Then $!_{n+1}$ is defined as $\alpha; \mathbf{F}(!_n)$. This function defines a new finer partition on X. If the partition is equivalent to that of $!_n$, then this partition equates all and only the bisimilar states (i.e., coincides with !).
   If the set of states is finite, then the algorithm terminates.



Unfortunately, due to cardinality reasons, the category of $\mathbf{P_L}$-coalgebras does not have a final object [15]. One satisfactory solution consists of 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 defining the functor $\mathbf{P_L^c} : \mathbf{Set} \to \mathbf{Set}$ by $X \mapsto \mathbf{P}_c(L \times X)$ one has that coalgebras for this endofunctor are one-to-one with transition systems with *countable degree*. Unlike functor $\mathbf{P_L}$, functor $\mathbf{P_L^c}$ admits final coalgebras (Ex. 6.8 of [15]).

The coalgebraic representation using functor $\mathbf{P_L^c}$ is not completely satisfactory, because by definition the carrier of a coalgebra is just a set and therefore the intrinsic algebraic structure of states is lost. This calls for the introduction of *structured coalgebras*, i.e., coalgebras for an endofuctor on a category $\mathbf{Alg_\Gamma}$ of algebras for a specification Γ. Since cohomomorphisms in a category of structured coalgebras are also Γ-homomorphisms bisimilarity is a congruence w.r.t. the operations in Γ.

In [19], bialgebras are used as structures combining algebras and coalgebras. Bialgebras are richer than structured coalgebras, since they can be seen both as coalgebras on algebras and also as algebras on coalgebras. Categories of bialgebras over the functor $\mathbf{P_L^c}$ have a final object and bisimilarity abstracts from the algebraic structure.

In [6], it is proved that whenever the endofunctor on algebras is a lifting of $\mathbf{P_L^c}$, then structured coalgebras coincide with bialgebras.

Proposition 3. *Let Γ be an algebraic specification. Let $\mathbf{V^\Gamma} : \mathbf{Alg_\Gamma} \to \mathbf{Set}$ be the forgetful functor. If $\mathbf{F_\Gamma} : \mathbf{Alg_\Gamma} \to \mathbf{Alg_\Gamma}$ is a* lifting of $\mathbf{P_L^c}$ along $\mathbf{V^\Gamma}$ *(i.e., $\mathbf{F_\Gamma}; \mathbf{V^\Gamma} = \mathbf{V^\Gamma}; \mathbf{P_L^c}$), then $\mathbf{F_\Gamma}$-coalgebras are bialgebras and $\mathbf{Coalg_{F_\Gamma}}$ has a final object.*

4 Coalgebraic Models of *CTS*s

In this section we give a coalgebraic characterization of Context Transition Systems of reactive systems through the theory outlined in the previous section. We will first define the *CTS* as a coalgebra without algebraic structure and then we will lift it to a structured setting. This proves that bisimilarity on *CTS* (i.e., $\sim_{SAT}$) is a congruence, and moreover it provides a characterization of $\sim_{SAT}$ as final semantics.

Firstly we have to define the universe of observations. Since the labels of the *CTS* are arrows of the base category $\mathbf{C}$ (representing the contexts), we define the functor as parametric w.r.t. $\mathbf{C}$, and $||\mathbf{C}||$ (i.e. the class of all arrows of $\mathbf{C}$) is the universe of labels.

Definition 7. *Given a category $\mathbf{C}$, the functor $\mathbf{P_C} : \mathbf{Set}^{|\mathbf{C}|} \to \mathbf{Set}^{|\mathbf{C}|}$ is defined for each $|\mathbf{C}|$-indexed set S by $\mathbf{P_C}(S_n) = \mathbf{P}_c\left(\bigcup_{m \in |\mathbf{C}|} \mathbf{C}[n, m] \times S_m\right)$.*
The functor is defined analogously on arrows of $\mathbf{Set}^{|\mathbf{C}|}$.

Note that $\mathbf{P_C}$ is not an endofunctor on $\mathbf{Set}$, as it is the case for the standard $\mathbf{P_L}$ discussed in the previous section, but it is defined on $\mathbf{Set}^{|\mathbf{C}|}$, i.e., the category of sets indexed by objects of $\mathbf{C}$. The base category $\mathbf{C}$ induces $\overline{\mathbf{C}}$, an object of $\mathbf{Set}^{|\mathbf{C}|}$ where, for any sort n, the corresponding set is $\mathbf{C}[0, n]$. Here we have implicitly assumed that $\mathbf{C}$ is *locally small* (i.e., the hom-class between two objects is always a set and not a proper class), otherwise $\mathbf{C}[0, n]$ could be a proper class. Moreover, in the following definition, we require that $||\mathbf{C}||$ is a countable set, otherwise the possible transitions of an element could beuncountable and then not belong to $\mathbf{P_C}$. Note that this usually holds in those categories where arrows are syntactic contexts of a formalism.

Definition 8. *Given a reactive system $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R}\rangle$, the $\mathbf{P_C}$-coalgebra corresponding to its CTS is $\langle \overline{\mathbf{C}}, \alpha_{\mathcal{R}}\rangle$ where $\alpha_{\mathcal{R}}(p) = \{(c, r; d)$ s.t. diagram (i) in Fig. 1 commutes and $d \in \mathbf{D}$ and $\langle l, r\rangle \in \mathbb{R}\}$.*

It is immediate to see that the LTS defined above exactly coincides with the *CTS* (Def. 2). However this model does not take into account the algebraic structure of the states, i.e., of the possibility of contextualizing a term. In order to have a richer model we lift this construction to a structured setting where the base category is not anymore $\mathbf{Set}^{|\mathbf{C}|}$, but a category of algebras with contextualization operations. In the following we assume that the category $\mathbf{C}$ has *strict distinguished object*, i.e., that the only arrow with target 0 is id_0. This is needed to distinguish between elements and operations of algebras.

specification $\Gamma(\mathcal{R})$ =
  sorts
    n $\forall n \in |\mathbf{C}|$ with $n \neq 0$
  operations
    $d : n \to m$ $\forall d \in \mathbf{C}[n, m]$ with $n \neq 0$
  equations
    $id(x) = x$
    $e(d(x)) = c(x)$ $\forall d; e = c$

This signature defines $\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ the category of $\Gamma(\mathcal{R})$-algebras. The base category $\mathbf{C}$ of a reactive system induces $\widehat{\mathbf{C}} \in |\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}|$. In $\widehat{\mathbf{C}}$, for every sort m, the elements of this sort are the arrows of $\mathbf{C}[0, m]$. Every operation $c : m \to n$ is defined for every element p of sort m as the composition of $p; c$ in $\mathbf{C}$.

Hereafter we will use $c_{\mathfrak{X}}$ to denote the operation c of the algebra $\mathfrak{X}$, and c to mean both the operation $c_{\widehat{\mathbf{C}}}$ and the arrow $c \in ||\mathbf{C}||$. Moreover we will not specify the sort of sets and operations, in order to make the whole presentation more readable.

Definition 9. *The functor $\mathbf{F} : \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})} \to \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ is defined as follows. For each $\mathfrak{X} = \langle X, a_{\mathfrak{X}}, b_{\mathfrak{X}}, \dots\rangle \in \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$, $\mathbf{F}(\mathfrak{X}) = \langle \mathbf{P_C}(X), a_{\mathbf{F}(\mathfrak{X})}, b_{\mathbf{F}(\mathfrak{X})}, \dots\rangle$ where $\forall a \in \Gamma(\mathcal{R})$, $\forall A \in \mathbf{P_C}(X)$, $a_{\mathbf{F}(\mathfrak{X})}(A) = \{(c, d_{\mathfrak{X}}(x))$ s.t. diagram (ii) in Fig. 1 commutes in $\mathbf{C}$, $d \in \mathbf{D}$ and $(b, x) \in A\}$. On arrows of $\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ is defined as $\mathbf{P_C}$.*

Trivially $\mathbf{F}$ is a lifting of $\mathbf{P_C}$. Then, by Prop. 3, $\mathbf{Coalg_F}$ is a category of bialgebras, it has final object $1_{\mathbf{Coalg_F}}$ and bisimilarity abstracts away from the algebraic structure.

In [19], Turi and Plotkin show that every process algebra whose operational semantics is given by SOS rules in DeSimone format, 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 SOS rules in DeSimone format specify how an endofunctor $\mathbf{F_\Sigma}$ behaves with respect to the operations of the signature. Since there exists only one arrow $?_\Sigma : T_\Sigma \to \mathbf{F_\Sigma}(T_\Sigma)$, to give the SOS rules is enough for defining a 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 $\widehat{\mathbf{C}}$, that is not the initial algebra of $\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$. Then there could exist several or no structured coalgebras with carrier $\widehat{\mathbf{C}}$. In the following we prove that $\alpha_{\mathcal{R}} : \widehat{\mathbf{C}} \to \mathbf{F}(\widehat{\mathbf{C}})$ is a $\Gamma(\mathcal{R})$-homomorphism. This automatically assures that $\langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}} \rangle$ is a structured coalgebra and then bisimilarity is a congruence with respect to the operations of $\Gamma(\mathcal{R})$.

Theorem 1. *Let $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$ be a reactive system. If $||\mathbf{C}||$ is countable and $\mathbf{C}$ has strict distinguished object, then $\langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}} \rangle$ is a $\mathbf{F}$-coalgebra.*

From the above theorem immediately follows the characterization of $\sim_{SAT}$ as final semantics. Indeed the unique cohomorphism $!_{\mathcal{R}} : \langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}} \rangle \to 1_{\mathbf{Coalg_F}}$ identifies all the bisimilar states of $\widehat{\mathbf{C}}$. In other words, for all $f, g \in ||\mathbf{C}||$ with domain 0, $f \sim_{SAT} g$ if and only if $!_{\mathcal{R}}(f) = !_{\mathcal{R}}(g)$.

5 Coalgebraic Models of *ITS*s

Analogously to the previous section, we define a $\mathbf{P_C}$-coalgebra that coincides with *ITS*.

Definition 10. *Given a reactive system $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$, the $\mathbf{P_C}$-coalgebra corresponding to its ITS is $\langle \bar{\mathbf{C}}, \alpha^I_{\mathcal{R}} \rangle$ where $\alpha^I_{\mathcal{R}}(p) = \{(c, r; d)$ s.t. diagram (i) in Fig. 1 is an IPO and $d \in \mathbf{D}$ and $\langle l, r \rangle \in \mathbb{R}\}$.*

Now we would like to lift this coalgebra to the structured setting of $\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$, but this is impossible, since $\alpha^I_{\mathcal{R}} : \widehat{\mathbf{C}} \to \mathbf{F}(\widehat{\mathbf{C}})$ is not a $\Gamma(\mathcal{R})$-homomorphism. Then we define below a different functor that is suitable for lifting $\alpha^I_{\mathcal{R}}$.

Definition 11. *The functor $\mathbf{I} : \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})} \to \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ is defined as follows.
For each $\mathfrak{X} = \langle X, a_{\mathfrak{X}}, b_{\mathfrak{X}}, \dots \rangle \in \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$, $\mathbf{I}(\mathfrak{X}) = \langle \mathbf{P_C}(X), a_{\mathbf{I}(\mathfrak{X})}, b_{\mathbf{I}(\mathfrak{X})}, \dots \rangle$
where $\forall a \in \Gamma(\mathcal{R})$, $\forall A \in \mathbf{P_C}(X)$, $a_{\mathbf{I}(\mathfrak{X})}(A) = \{(c, d_{\mathfrak{X}}(x))$ s.t. diagram (ii) in Fig. 1 is an IPO in $\mathbf{C}$ and $(b, x) \in A$ and $d \in \mathbf{D}\}$. On arrows of $\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ is defined as $\mathbf{P_C}$.*

Trivially, also $\mathbf{I}$ is a lifting of $\mathbf{P_C}$. The following theorem assures that $\sim_{IPO}$ is a congruence, and gives us a characterization as final semantics.

Theorem 2. *Let $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$ be a reactive system with redex-RPOs. If $||\mathbf{C}||$ is a countable and $\mathbf{C}$ has strict distinguished object, then $\langle \widehat{\mathbf{C}}, \alpha^I_{\mathcal{R}} \rangle$ is an $\mathbf{I}$-coalgebra.*

It is worth to note that the existence of redex-RPOs is fundamental in order to prove that $\alpha^I_{\mathcal{R}} : \widehat{\mathbf{C}} \to \mathbf{I}(\widehat{\mathbf{C}})$ is a $\Gamma(\mathcal{R})$-homomorphism, while it is not necessary for $\alpha_{\mathcal{R}}$.

A different coalgebraic construction for *ITS* has been already proposed in [5].

6 Normalized Coalgebras

The coalgebraic characterization of $\sim_{SAT}$ given in Sec. 4, is not completely satisfactory. While it supplies a characterization as final semantics, it does not allow for a minimization procedure because *CTS* is usually infinitely branching. Similar motivations have driven us to introduce semi-saturated bisimilarity in [4], that efficiently characterizes saturated bisimilarity. In this section we use the main intuition of semi-saturated bisimilarity in order to give an efficient and coalgebraic characterization of $\sim_{SAT}$. We introduce $\mathbf{Coalg_N}$, the category of Normalized Coalgebras, and we prove that it is isomorphic to $\mathbf{Coalg_F}$ (Sec. 6.1). This allows us to characterizes $\sim_{SAT}$ as the final morphism in $\mathbf{Coalg_N}$. Sec. 6.2 shows that, in the finite case, the coalgebraic minimization algorithm in $\mathbf{Coalg_N}$ is computable and Ex. 2 applies it to the open net $\mathcal{N}$.

Recall the definition of semi-saturated bisimulation (Def. 5). When p proposes a move labeled by a context c, then q must answer with a move labeled by the same context, or by a smaller one. Suppose that it answers with a smaller context d. Since bisimulations are symmetric, q will propose the d move and now p must perform a transition labeled by d, or by a smaller context. Our intuition is that, if the category of contexts is in some sense well formed, and if p and q are bisimilar, at the end p and q must perform both a transition labeled with the same minimal context. All the other bigger transitions are *redundant*, i.e., meaningless in the bisimulation game.

Thus, in order to capture the right bisimilarity, we have to forget about all the redundant transitions, i.e., all transitions $p \xrightarrow{c} p'$ such that $p \xrightarrow{d} p''$ and $\exists e \in \mathbf{D}$ with $c = d; e$ and $p' \sim p''; e$. As an example consider the *ITS* of the Open Petri net $\mathcal{N}$ (Fig. 2). The transition $e \xrightarrow{y} fy$ is redundant because $e \xrightarrow{id} f$ and clearly $fy \sim fy$ (note that in our example $m; n = mn$ for all multisets m and n). The transition $a \xrightarrow{xy} e$ is redundant because $a \xrightarrow{y} c$ and $cx \sim e$ (proved in Ex. 1).

But immediately a problem arises. How can we decide which transitions are redundant, if redundancy itself depends on bisimilarity?

Our proposal is the following. First we consider redundant only the transitions $p \xrightarrow{c} p'$ such that $p \xrightarrow{d} p''$ and $p' = p''; e$ (where as usual $c = d; e$). In our example $e \xrightarrow{y} fy$ is redundant, while $a \xrightarrow{xy} e$ is not. Then we define a category of coalgebras without redundant transitions. Since in the final object, all the bisimilar states are identified, all the transitions $p \xrightarrow{c} p'$ such that $p \xrightarrow{d} p''$ and $p' \sim p''; e$ will be forgotten.

We can better understand this idea thinking about minimization. We *normalize*, i.e., we junk away all the redundant transitions (those where $p' = p''; e$) and then we minimize w.r.t. bisimilarity. Now the bisimilar states are identified and if we normalize again, we will junk away new redundant transitions. We repeat this procedure until we reach a fix point (the final object). Since all the bisimilar states are identified in the final object, we will have forgotten not only all the transitions $p \xrightarrow{c} p'$ such that $p \xrightarrow{d} p''$ and $p' = p''; e$, but also those where $p' \sim p''; e$.

Consider for example the *ITS* derived from $\mathcal{N}$ (Fig. 2(ii)). After normalization the transition $e \xrightarrow{y} fy$ disappears (Fig. 3(i)) and after minimization $e = cx$ (Fig. 3(ii)). If we normalize again we also junk away the transition $a \xrightarrow{xy} e$ and performing a further minimization we reach the LTS depicted in Fig. 3(iii).

It is worth to note that normalization and minimization have to be repeated iteratively. Indeed we cannot minimize once and then normalize, or normalize once and then minimize (try with our example).

Definition 12 (Normalized Set and Normalization). *Let $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$ be a reactive system. Let $\mathfrak{X}$ be a $\Gamma(\mathcal{R})$-algebra with carrier set X and $A \in \mathbf{P_C}(X)$.*

A transition (c', x') derives *(c, x) in $\mathfrak{X}$ (in symbols $(c', x') \vdash_{\mathfrak{X}} (c, x)$) iff $\exists d \in \mathbf{D}$ such that $c'; d = c$ and $d_{\mathfrak{X}}(x') = x$. A transition (c', x') is* equivalent *to (c, x) in $\mathfrak{X}$*

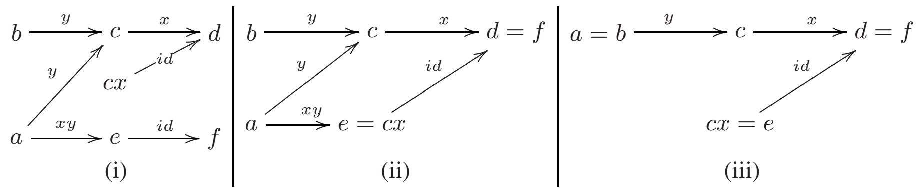


Fig. 3. (i) The portion of $\langle\widehat{\mathbf{OPL}}, \alpha^I_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}\rangle$ corresponding to a and b.(ii) $\langle\mathfrak{B}, \beta\rangle$ is not a $\mathbf{N}$-coalgebra. (iii)$\langle\mathfrak{C}, \gamma\rangle$ is a $\mathbf{N}$-coalgebra.

($(c', x') \equiv_{\mathfrak{X}} (c, x)$) iff $(c', x') \vdash_{\mathfrak{X}} (c, x)$ and $(c, x) \vdash_{\mathfrak{X}} (c', x')$. A transition (c', x') dominates *(c, x) in $\mathfrak{X}$ ($(c', x') \prec_{\mathfrak{X}} (c, x)$) iff $(c', x') \vdash_{\mathfrak{X}} (c, x)$ and $(c, x) \nvdash_{\mathfrak{X}} (c', x')$. A transition $(c, x) \in A$ is said* redundant in *A w.r.t. $\mathfrak{X}$ if $\exists (c', x') \in A$ such that $(c', x') \prec_{\mathfrak{X}} (c, x)$.*

A is normalized *in $\mathfrak{X}$ iff it does not contain redundant transitions and it is closed by equivalent transitions. The set $\mathbf{P}^{\mathbf{N}\mathfrak{X}}_{\mathbf{C}}(X)$ is the subset of $\mathbf{P}_{\mathbf{C}}(X)$ containing all and only the normalized sets in $\mathfrak{X}$.*

For any $A \in \mathbf{P}_{\mathbf{C}}(X)$, the normalization function *$norm_{\mathfrak{X}} : \mathbf{P}_{\mathbf{C}}(X) \to \mathbf{P}^{\mathbf{N}\mathfrak{X}}_{\mathbf{C}}(X)$ maps $A \in \mathbf{P}_{\mathbf{C}}(X)$ in $\{(c', l')$ s.t. $(c', l') \equiv (c, l) \in A$ and (c, l) not redundant in A w.r.t. $\mathfrak{X}\}$.*

Look at the *ITS* of $\mathcal{N}$ in Fig.2(ii). The set of IPO transitions of e (i.e., $\alpha^I_{\mathcal{N}}(e)$) is not normalized in $\widehat{\mathbf{OPL}}$, because $(id, f) \prec_{\widehat{\mathbf{OPL}}} (x, fx)$, while the set of IPO transitions of a is normalized since $(x, c) \nvdash_{\widehat{\mathbf{OPL}}} (xy, e)$ because $y_{\widehat{\mathbf{OPL}}}(c) \neq e$. (Remember that $\mathbf{OPL}$ is the base category of $\mathcal{N}$. The algebra $\widehat{\mathbf{OPL}}$ can be thought roughly as an algebra having multisets as both elements and operators where $\forall m, n$ multisets, $m(n) = m \oplus n$ where $\oplus$ is the union of multisets).

Normalizing a set of transitions means eliminating all the redundant transitions and then closing w.r.t. $\equiv$. It is worth to note that we use $\prec_{\mathfrak{X}}$ (and not $\vdash_{\mathfrak{X}}$) to define redundant transitions. Indeed, suppose that $(c, x) \equiv (c', x')$ and no other transition dominates them. If we consider both redundant, then normalization erases both of them. This is in contrast with our main intuition of normalization, i.e., the normalized set must contain all the minimal transitions needed to derive the original set (Lemma 1).

Definition 13 (Normalizable Reactive System). *A Reactive System $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R}\rangle$ is normalizable if:*

1. *$||\mathbf{C}||$ is countable,*
2. *$\mathbf{C}$ has strict distinguished object,*
3. *$\forall \mathfrak{X} \in \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$, $\prec_{\mathfrak{X}}$ is well founded.*

We inherit the first and the second constraint by Sec. 4. The third assures that for any transition, there exists a minimal non redundant transition that dominates it.

Lemma 1. *Let $\mathcal{R}$ be a normalizable reactive system. Let $\mathfrak{X}$ be $\Gamma(\mathcal{R})$-algebra and $A \in |\mathbf{F}(\mathfrak{X})|$. Then $\forall (d, x) \in A$, $\exists (d', x') \in norm_{\mathfrak{X}}(A)$, such that $(d', x') \prec_{\mathfrak{X}} (d, x)$.*

We cannot prove that the third constraint is less restrictive than requiring to have redex-RPOs, but while the latter usually does not hold in categories representing syntactic contexts (look at Ex. 2.2.2. of [18]), the former will usually hold. Indeed it just requires that a context cannot be decomposed infinitely many times.

Definition 14. *The functor* $\mathbf{N} : \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})} \to \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ *is defined as follows. For each* $\mathfrak{X} = \langle X, a_{\mathfrak{X}}, b_{\mathfrak{X}}, \dots \rangle$, $\mathbf{N}(\mathfrak{X}) = \langle \mathbf{P}^{\mathbf{N}\mathfrak{X}}_{\mathbf{C}}(X), a_{\mathbf{F}(\mathfrak{X})}; norm_{\mathfrak{X}}, b_{\mathbf{F}(\mathfrak{X})}; norm_{\mathfrak{X}}, \dots \rangle$.
For all $h : \mathfrak{X} \to \mathfrak{Y}$, $\mathbf{N}(h) = \mathbf{F}(h); norm_{\mathfrak{Y}}$.

The $\mathbf{I}$-coalgebra corresponding to $\mathcal{N}$, namely $\langle \widehat{\mathbf{OPL}}, \alpha^I_{\mathcal{N}} \rangle$ (partially in Fig.2(ii)), is not a $\mathbf{N}$-coalgebra since $\alpha^I_{\mathcal{N}}(e)$ is not normalized in $\widehat{\mathbf{OPL}}$. On the other hand, it is easy too see that $\langle \widehat{\mathbf{OPL}}, \alpha^I_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}} \rangle$ (partially represented in Fig.3(i)) is a $\mathbf{N}$-coalgebra.

Note how the functor is defined on arrows. If we apply $\mathbf{F}(h)$ to a normalized set A, the resulting set may not be normalized. Thus we apply the normalization function $norm_{\mathfrak{Y}}$, after the mapping $\mathbf{F}(h)$.

This is the most important intuition behind normalized coalgebras. Normalization after mapping makes bisimilar also transition systems which are such only forgetting redundant transitions. Let $\mathfrak{C}$ be the algebra obtained by quotienting $\widehat{\mathbf{OPL}}$ with $e = xc$, $a = b$ and $d = f$ and let h be such a quotient. Let γ be the transition structure on $\mathfrak{C}$ partially represented in Fig.3(iii). We have that $\alpha^I_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}; \mathbf{F}(h) \neq h; \gamma$, since $\alpha^I_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}; \mathbf{F}(h)(a) = \{(xy, e), (y, c)\}$ and $h; \gamma(a) = \{(y, c)\}$. But $\alpha^I_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}; \mathbf{F}(h); norm_{\mathfrak{Y}} = h; \gamma$ ($\alpha^I_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}; \mathbf{F}(h); norm_{\mathfrak{Y}}(a) = \{(y, c)\}$).

Now we would like to apply the theory illustrated in Sec. 3, as we have done for $\mathbf{F}$ and $\mathbf{I}$, but this is impossible since the notion of normalization (and hence the functor) strictly depends on the algebraic structure. In categorical terms, this means that $\mathbf{N}$-coalgebras are not bialgebras, or equivalently, that there exists no functor $\mathbf{B} : \mathbf{Set}^{|\mathbf{S}|} \to \mathbf{Set}^{|\mathbf{S}|}$ such that $\mathbf{N}$ is a lifting of $\mathbf{B}$.

6.1 Isomorphism Theorem

Here we prove that $\mathbf{Coalg}_{\mathbf{F}}$ and $\mathbf{Coalg}_{\mathbf{N}}$ are isomorphic. This assures that $\mathbf{Coalg}_{\mathbf{N}}$ has a final object. Moreover the final morphism in $\mathbf{Coalg}_{\mathbf{N}}$ still characterizes $\sim_{SAT}$.

We start by introducing a new category of coalgebras that is isomorphic to both $\mathbf{Coalg}_{\mathbf{F}}$ and $\mathbf{Coalg}_{\mathbf{N}}$.

Definition 15. *Let* $\mathcal{R}$ *be a reactive system and* $\mathfrak{X}$ *be a* $\Gamma(\mathcal{R})$*-algebra with carrier set* X. *A set* $A \in \mathbf{P_C}(X)$ *is* saturated *in* $\mathfrak{X}$ *if and only if it is closed w.r.t.* $\vdash_{\mathfrak{X}}$. *The set* $\mathbf{P}^{\mathbf{S}\mathfrak{X}}_{\mathbf{C}}(X)$ *is the subset of* $\mathbf{P_C}(X)$ *containing all and only the saturated sets in* $\mathfrak{X}$.

For any $A \in \mathbf{P_C}(X)$, *the* saturation function $sat_{\mathfrak{X}} : \mathbf{P_C}(X) \to \mathbf{P}^{\mathbf{S}\mathfrak{X}}_{\mathbf{C}}(X)$ *maps* A *to* $\{(c', x')$ *s.t.* $(c, x) \in A$ *and* $(c, x) \vdash_{\mathfrak{X}} (c', x')\}$.

The functor $\mathbf{S} : \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})} \to \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ *is defined as follows.*
For each $\mathfrak{X} = \langle X, a_{\mathfrak{X}}, b_{\mathfrak{X}}, \dots \rangle \in \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$, $\mathbf{S}(\mathfrak{X}) = \langle \mathbf{P}^{\mathbf{S}\mathfrak{X}}_{\mathbf{C}}(X), a_{\mathbf{F}(\mathfrak{X})}, b_{\mathbf{F}(\mathfrak{X})}, \dots \rangle$.
On arrows of $\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$ *is defined as* $\mathbf{F}$.

The only difference between $\mathbf{S}$ and $\mathbf{F}$ is that for any $\Gamma(\mathcal{R})$-algebra $\mathfrak{X}$, $|\mathbf{S}(\mathfrak{X})|$ contains only the saturated set of transitions, while $|\mathbf{F}(\mathfrak{X})|$ contains all the possible sets of transitions. In terms of coalgebras this means that the coalgebras of the former functor are forced to perform only saturated set of transitions, while coalgebras of the latter are not. But every $\mathbf{F}$-coalgebra $\langle \mathfrak{X}, \alpha \rangle$ is however forced to performs only saturated set of transitions. Indeed $\forall x \in |\mathfrak{X}|$, $\alpha(x) = \alpha(id_{\mathfrak{X}}(x)) = id_{\mathbf{F}(\mathfrak{X})}(\alpha(x))$ because α is a homomorphism. By definition $id_{\mathbf{F}(\mathfrak{X})}(\alpha(x))$ is closed w.r.t. $\vdash_{\mathfrak{X}}$, i.e., saturated.

The left triangle of diagram (i) in Fig.4 depicts this setting. $\mathbf{S}(\mathfrak{X})$ is a subalgebra of $\mathbf{F}(\mathfrak{X})$, i.e. $\forall \mathfrak{X} \in \mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}, \quad \exists m_{\mathfrak{X}} : \mathbf{S}(\mathfrak{X}) \rightarrowtail \mathbf{F}(\mathfrak{X})$ mono. Moreover $\forall \alpha : \mathfrak{X} \to \mathbf{F}(\mathfrak{X})$, there exists a unique $\alpha_S : \mathfrak{X} \to \mathbf{S}(\mathfrak{X})$ such that $\alpha = \alpha_S; m_{\mathfrak{X}}$. This observation guarantees that $\mathbf{Coalg_F}$ and $\mathbf{Coalg_S}$ are isomorphic. While, in order to prove the isomorphism of $\mathbf{Coalg_S}$ and $\mathbf{Coalg_N}$, we show that normalization and saturation are natural isomorphisms.

Proposition 4. *Let* $norm$ *and* sat *be respectively the families of morphisms* $\{norm_{\mathfrak{X}} : \mathbf{S}(\mathfrak{X}) \to \mathbf{N}(\mathfrak{X}) \ \forall \mathfrak{X} \in |\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}|\}$ *and* $\{sat_{\mathfrak{X}} : \mathbf{N}(\mathfrak{X}) \to \mathbf{S}(\mathfrak{X}) \ \forall \mathfrak{X} \in |\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}|\}$. *Then* $norm : \mathbf{S} \Rightarrow \mathbf{N}$ *and* $sat : \mathbf{N} \Rightarrow \mathbf{S}$ *are natural transformations, and one is the inverse of the other, i.e. all squares in diagram (ii) of Fig.4 commutes.*

Theorem 3. $\mathbf{Coalg_F}$, $\mathbf{Coalg_S}$ *and* $\mathbf{Coalg_N}$ *are isomorphic.*

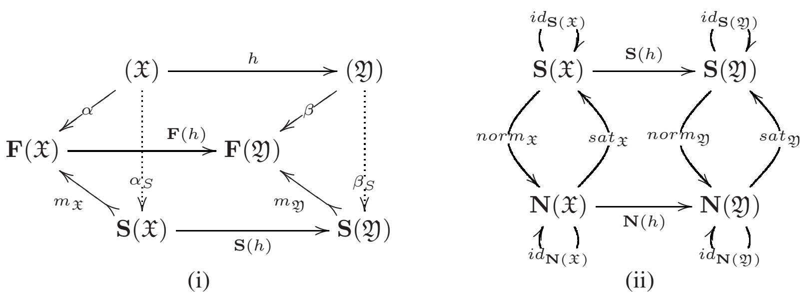


Fig. 4. Commuting diagrams in $\mathbf{Alg}_{\mathbf{\Gamma}(\mathcal{R})}$

The above theorem guarantees that $\mathbf{Coalg_N}$ has a final system $1_{\mathbf{Coalg_N}}$. The final morphisms $!^N_{\mathcal{R}} : \langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}} \rangle \to 1_{\mathbf{Coalg_N}}$ characterizes $\sim_{SAT}$.

Corollary 1. *Let* $\mathcal{R}$ *be a normalizable reactive system.* $p \sim_{SAT} q \Leftrightarrow !_{\mathcal{R}}(p) = !_{\mathcal{R}}(q) \Leftrightarrow !^N_{\mathcal{R}}(p) = !^N_{\mathcal{R}}(q)$.

6.2 From *ITS* to $\sim_{SAT}$ Through Normalization

Until now, we have proved that $!^N_{\mathcal{R}} : \langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}} \rangle \to 1_{\mathbf{Coalg_N}}$ characterizes $\sim_{SAT}$. Now we apply the coalgebraic minimization algorithm (Sec. 3) in the category $\mathbf{Coalg_N}$, in order to compute $\sim_{SAT}$. However, normalizing $\alpha_{\mathcal{R}}$ is unfeasible, because it is usually infinitely branching. Instead of normalizing the *CTS*, we can build $\alpha_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}}$ through the normalization of *ITS*.

Note that the *ITS* could have redundant transitions. Indeed consider two redex squares for two different rules as those depicted in diagram (vi) of Fig. 1 where $\langle l, r \rangle$,

$\langle l', r' \rangle \in \mathbb{R}$. The transition $p \xrightarrow{c} r; d$ could be an IPO transition even if it is dominated by $p \xrightarrow{c'} r'; d'$. This explains the difference between our notion of redundancy and that of Leifer and Milner. They consider all the non-IPO transitions redundant, i.e. all the transitions where the label contains something that is not strictly necessary to reach the rule. Our notion completely abstracts from rules.

Theorem 4. *Let $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$ be a normalizable reactive system having IPOs. Then $\alpha^I_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}} = \alpha_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}}$ and moreover, $a_{\mathbf{I}(\mathfrak{X})}; norm_{\mathfrak{X}} = a_{\mathbf{F}(\mathfrak{X})}; norm_{\mathfrak{X}}$.*

This theorem is the key to compute $!^N_{\mathcal{R}}(p)$. Indeed it allows to compute $\alpha^I_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}}$ instead of $\alpha_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}}$ that is usually unfeasible. Now we can instantiate the general minimization algorithm (Sec. 3) in the case of $\mathbf{Coalg_N}$.

At the beginning $!^N_0 : \widehat{\mathbf{C}} \to \mathfrak{1}$ is the final morphism to $\mathfrak{1}$ (the final $\Gamma(\mathcal{R})$-algebra).

At any iteration $!^N_{n+1} = \alpha_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}}; \mathbf{N}(!^N_n) = \alpha^I_{\mathcal{R}}; norm_{\widehat{\mathbf{C}}}; \mathbf{F}(!^N_n); norm_{\mathbf{N}(\mathfrak{1})^n}$.

The peculiarity of minimization in $\mathbf{Coalg_N}$ is that we must normalize at every iteration. Note that the normalization is performed not only in the source algebra $\widehat{\mathbf{C}}$, but also on the target algebra $\mathbf{N}(\mathfrak{1})^n$. Thus the minimization procedure strictly depends on the algebraic structure. This further explains why normalized coalgebras are structured coalgebras but not bialgebras where we can completely forget about the algebraic structure.

Proposition 5. *Let $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$ be a normalizable reactive system such that:*

- *arrow composition in $\mathbf{C}$ is computable,*
- *$\mathbf{C}$ has IPOs, and these can be computed,*
- *$\forall a, b \in ||\mathbf{C}||$, there exist a finite number of $c \in ||\mathbf{C}||$ such that $a = b; c$,*
- *$\forall a, b \in ||\mathbf{C}||$, there exist a finite number of $c, d \in ||\mathbf{C}||$ such that diagram (ii) in Fig. 1 is an IPO.*

Then the algorithm outlined above is computable and it terminates for minimizing those p whose ITS is finite.

Example 2. Here we prove that $a \sim_{SAT} b$ in $\mathcal{N}$ (Ex. 1), by proving that $!^N_{\mathcal{N}}(a) = !^N_{\mathcal{N}}(b)$. The LTSs $\alpha_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}(a)$ and $\alpha_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}(b)$ are shown in Fig.3(i) (by Th. 4 these can be computed by normalizing in $\widehat{\mathbf{OPL}}$ the *ITS* in Fig. 2(ii)). Let $\Gamma(\mathcal{N})$ be the specification corresponding to $\mathcal{N}$: operations are just multisets on $\{x, y\}$. Let $\mathfrak{1}$ be the final $\Gamma(\mathcal{N})$-algebra: the carrier set contains only the single element 1 and for all operations m, $m(1) = 1$.

The homomorphism $!^N_0 : \widehat{\mathbf{OPL}} \to \mathfrak{1}$ maps all the elements of $|\widehat{\mathbf{OPL}}|$ into 1.

In order to compute $!^N_1$, we first compute $\alpha_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}; \mathbf{F}(!^0)$ for all the states reachable from a and b (the results are reported in the second column of Fig. 5(i)) and then we normalize in the final algebra $\mathfrak{1}$ (third column). The normalization junks away the transition $a \xrightarrow{xy} 1$. Indeed $(y, 1) \prec_{\mathfrak{1}} (xy, 1)$ since $xy = y; x$ and $x_{\mathfrak{1}}(1) = 1$.

For computing $!^N_2$ we proceed as before, using $!^N_1$ instead of $!^N_0$ and normalizing on $\mathbf{N}(\mathfrak{1})$ instead of normalizing on $\mathfrak{1}$. The results of the second iteration are reported in Fig. 5(ii). Normalization junks away the transitions $a \xrightarrow{xy} \{(id, 1)\}$ because $(y, \{(x, 1)\}) \prec_{\mathbf{N}(\mathfrak{1})} (xy, \{(id, 1)\})$. The morphism $!^N_2$ partitions the states in $\{a, b\}$, $\{c\}$, $\{d, f\}$, $\{e\}$, as well as $!^N_1$. Thus the algorithm terminates and then $a \sim_{SAT} b$.

multisets	$\alpha_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}; \mathbf{F}(!_0^N)$	$!_1^N$
a	$(xy, 1), (y, 1)$	$(y, 1)$
b	$(y, 1)$	$(y, 1)$
c	$(x, 1)$	$(x, 1)$
d	$\varnothing$	$\varnothing$
e	$(id, 1)$	$(id, 1)$
f	$\varnothing$	$\varnothing$

(i)

multisets	$\alpha_{\mathcal{N}}; norm_{\widehat{\mathbf{OPL}}}; \mathbf{F}(!_1^N)$	$!_2^N$
a	$(xy, \{(id, 1)\}), (y, \{(x, 1)\})$	$(y, \{(x, 1)\})$
b	$(y, \{(x, 1)\})$	$(y, \{(x, 1)\})$
c	$(x, \varnothing)$	$(x, \varnothing)$
d	$\varnothing$	$\varnothing$
e	$(id, \varnothing)$	$(id, \varnothing)$
f	$\varnothing$	$\varnothing$

(ii)

Fig. 5. (i) First Iteration. (ii)Second Iteration.

7 Conclusions

In this paper we have defined structured coalgebras for both labeled transition systems (*CTS* and *ITS*), derived from reactive systems. This provides a characterization of $\sim_{SAT}$ and $\sim_{IPO}$ via final semantics and minimal realizations. Since *CTS* is usually infinite branching, its minimal realization is infinite and $\sim_{SAT}$ uncomputable. For this reason, we have introduced Normalized Coalgebras. These are structured coalgebras that, thanks to a suitable definition of the underlying functor, allow to forget about redundant transitions but still characterize $\sim_{SAT}$ as final semantics. Here the notion of redundancy is coarser than that expressed by the IPO condition. Indeed, given $p \xrightarrow{y} q$, $p \xrightarrow{x} q'$ and a context $E[-]$, such that $E[x] = y$, the former transition is not an IPO if the latter reacts with the same rule and $E[q'] = q$, while it is redundant, according to our notion of normalized coalgebras, if $E[q'] \sim q$ (without any condition on rules). Then constructing an LTS smaller than *ITS* and then minimizing it through a minimization algorithm (which employs the proper functor definition) allows us to check $\sim_{SAT}$. This approach can be easily extended to G-reactive systems [17] and to open reactive systems [11] where, in our opinion, it might help to relax the constraints of the theory. Pragmatically, Normalized Coalgebras are isomorphic to a category of bialgebras, but the minimization procedure is feasible, because it employs the algebraic structure that is completely forgotten in bialgebras.

References

1. Adámek, J., Koubek, V.: On the greatest fixed point of a set functor. Theoretical Computer Science 150(1) (1995)
2. Baldan, P., Corradini, A., Ehrig, H., Heckel, R.: Compositional semantics for open Petri nets based on deterministic processes. Mathematical Structures in Computer Science 15(1), 1–35 (2005)
3. Berry, G., Boudol, G.: The chemical abstract machine. Theoretical Computer Science 96, 217–248 (1992)
4. Bonchi, F., König, B., Montanari, U.: Saturated semantics for reactive systems. In: Proc. of LICS, pp. 69–80. IEEE Computer Society Press, Los Alamitos (2006)
5. Bonchi, F., Montanari, U.: A coalgebraic theory of reactive systems. Electronic Notes in Theoretical Computer Science, LIX Colloqium on Emerging Trends in Concurrency Theory (to appear)

6. Corradini, A., Große-Rhode, M., Heckel, R.: A coalgebraic presentation of structured transition systems. Theoretical Computer Science 260, 27–55 (2001)
7. Corradini, A., Heckel, R., Montanari, U.: From sos specifications to structured coalgebras: How to make bisimulation a congruence. Electronic Notes in Theoretical Computer Science, vol. 19 (1999)
8. Ehrig, H., König, B.: Deriving bisimulation congruences in the DPO approach to graph rewriting. In: Sassone, V. (ed.) FOSSACS 2005. LNCS, vol. 3441, pp. 151–166. Springer, Heidelberg (2005)
9. Kanellakis, P.C., Smolka, S.A.: Ccs expressions, finite state processes, and three problems of equivalence. Information and Computation 86(1), 43–68 (1990)
10. Kindler, E.: A compositional partial order semantics for Petri net components. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 235–252. Springer, Heidelberg (1997)
11. Klin, B., Sassone, V., Sobocinski, P.: Labels from reductions: Towards a general theory. In: Fiadeiro, J.L., Harman, N., Roggenbach, M., Rutten, J. (eds.) CALCO 2005. LNCS, vol. 3629, pp. 30–50. Springer, Heidelberg (2005)
12. Leifer, J.J., Milner, R.: Deriving bisimulation congruences for reactive systems. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 243–258. Springer, Heidelberg (2000)
13. Milner, R.: Bigraphs for Petri nets. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) Lectures on Concurrency and Petri Nets. LNCS, vol. 3098, pp. 686–701. Springer, Heidelberg (2004)
14. Milner, R.: Bigraphical reactive systems. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 16–35. Springer, Heidelberg (2001)
15. Rutten, J.J.M.M.: Universal coalgebra: a theory of systems. Theoretical Computer Science 249(1), 3–80 (2000)
16. Sassone, V., Sobociński, P.: A congruence for Petri nets. In: Ehrig, H., Padberg, J., Rozenberg, G. (eds.) Petri Nets and Graph Transformation. ENTCS, vol. 127, pp. 107–120. Elsevier, Amsterdam (2005)
17. Sassone, V., Sobociński, P.: Reactive systems over cospans. In: Proc. of LICS, pp. 311–320. IEEE Computer Society Press, Los Alamitos (2005)
18. Sobociński, P.: Deriving process congruences from reaction rules. PhD thesis (2004)
19. Turi, D., Plotkin, G.D.: Towards a mathematical operational semantics. In: Proc. of LICS, pp. 280–291. IEEE Computer Society Press, Los Alamitos (1997)